TESIS – SS09 2304

COMBINE SAMPLING – LEAST SQUARE SUPPORT VECTOR MACHINE UNTUK KLASIFIKASI MULTI CLASS IMBALANCED DATA

Hani Khaulasari
NRP. 1314201044

DOSEN PEMBIMBING
Santi Wulan Purnami, M.Si, Ph.D
Dr. rer. pol. Dedy Dwi Prastyo, M.Si

PROGRAM MAGISTER JURUSAN STATISTIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

THESIS – SS09 2304

COMBINE SAMPLING - LEAST SQUARE SUPPORT VECTOR MACHINE FOR MULTI CLASS IMBALANCED DATA CLASSIFICATION

Hani Khaulasari
NRP. 1314201044

SUPERVISOR
Santi Wulan Purnami, M.Si, Ph.D
Dr. rer. pol. Dedy Dwi Prastyo, M.Si

MAGISTER PROGRAM
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS AND NATURAL SCIENCES
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2016

COMBINE SAMPLING – LEAST SQUARE SUPPORT VECTOR MACHINE UNTUK KLASIFIKASI MULTI CLASS IMBALANCED DATA

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Master Sains (M.Si)
di
Institut Teknologi Sepuluh Nopember

oleh :
HANI KHAULASARI
NRP. 1314 201 044

Tanggal Ujian : 25 Januari 2015
Periode Wisuda : Maret 2016

Disetujui oleh:

1. **Santi Wulan Purnami, M.Si, Ph.D** (Pembimbing I)
   NIP: 19720923 199803 2 001

2. **Dr. rer.pol. Dedy Dwi Prastyo, M.Si** (Pembimbing II)
   NIP: 19831204 200812 1 002

3. **Dr. Purhadi, M.Sc** (Penguji)
   NIP: 19620204 198701 1 001

4. **Santi Puteri Rahayu, M.Si, Ph.D** (Penguji)
   NIP: 19750115 199903 2 003

Direktur Program Pascasarjana ITS,

Prof. Ir. Djauhar Manfaat, M. Sc., Ph. D.
NIP. 19601202 198701 1 001

COMBINE SAMPLING - LEAST SQUARE SUPPORT VECTOR MACHINE UNTUK KLASIFIKASI MULTI CLASS IMBALANCED DATA

Nama Mahasiswa : Hani Khaulsari
NRP : 1314201044
Pembimbing : Santi Wulan Purnami, M.Si, Ph.D
Co-Pembimbing : Dr. rer.pol. Dedy Dwi Prastyo, M.Si

ABSTRAK

Analisis Klasifikasi adalah proses menemukan model terbaik dari *classifier* untuk memprediksi kelas dari suatu objek atau data yang label kelasnya tidak diketahui. Pada kehidupan nyata, khususnya di bidang medis sering kali ditemui klasifikasi *multi class* dengan kondisi himpunan data *imbalanced*. Kondisi *imbalanced* data menjadi masalah dalam klasifikasi *multi class* karena mesin *classifier learning* akan condong memprediksi ke kelas data yang banyak (mayoritas) dibanding dengan kelas minoritas. Akibatnya, dihasilkan akurasi prediksi yang baik terhadap kelas data *training* yang banyak (kelas mayoritas) sedangkan untuk kelas data *training* yang sedikit (kelas minoritas) akan dihasilkan akurasi prediksi yang buruk. Oleh Karena itu, pada penelitian ini akan diterapkan metode *Combine Sampling (SMOTE+Tomek Links)* LS-SVM untuk klasifikasi *multi class imbalanced* dengan menggunakan data medis. Data yang digunakan adalah data thyroid, kanker payudara dan kanker serviks. Percobaan tersebut menggunakan q-fold cross validation (q=5) dan (q=10). LS-SVM *One Againt One* (OAO) digunakan untuk klasifikasi *multi class*. Optimasi parameter fungsi kernel RBF σ dan C menggunakan PSO-GSA Hasil menunjukan bahwa metode yang terbaik untuk digunakan dalam memprediksi status pasien penderita Thyroid, kanker payudara dan kanker serviks adalah metode *Combine Sampling Least Square Support Vector Machine PSO-GS*A. Klasifikasi dengan menggunakan Q-Fold (q=5) dan (q=10) menghasilkan performansi yang sama dalam hal akurasi Total, *Sensitivity* dan *G-Mean*.

Kata Kunci : *Imbalanced Data,* Klasifikasi *Multi Class,* LS-SVM*, SMOTE, Tomek Links, Combine Sampling, PSO-GSA.*

COMBINE SAMPLING LEAST SQUARE SUPPORT VECTOR MACHINE FOR MULTI CLASS IMBALANCED DATA CLASSIFICATION

Name : **Hani Khaulsari**
Student Id Number : **1314201044**
Supervisor : **Santi Wulan Purnami, M.Si, Ph.D**
Co-Supervisor : **Dr. rer.pol. Dedy Dwi Prastyo, M.Si**

ABSTRACT

Classification analysis is the process of finding the best model of a classifier for predicting the class of an object or data class label is unknown. In the real life, especially in the medical field often encountered multi-class classification with imbalanced data sets conditions. Imbalanced condition of the data at issue in multi-class classification as machine learning classifier will be inclined to predict that a lot of data classes (the majority) compared with a minority class. As a result, generated a good prediction accuracy of the data class training that many (the majority class), while for class training data bit (the minority) will produce a poor prediction accuracy. Hence, this research will apply the method Combine Sampling (SMOTE + Tomek Links) LS-SVM for multi-class classification imbalanced using medical data. The data used is data thyroid, breast cancer and cervical cancer. The experiment using a q-fold cross validation (q = 5) and (q = 10). LS-SVM One againt One (OAO) is used for multi-class classification. Parameter optimization RBF kernel function (σ) and C using the PSO-GSA. Results showed that the best method to use in predicting the status of patients with thyroid, breast cancer and cervical cancer is the combine Sampling method Least Square Support Vector Machine PSO-GSA. Classification by using Q-Fold (q = 5) and (q = 10) produces the same performance in terms of total accuracy, sensitivity and G-mean.

Keywords: Imbalanced Data , LS-SVM, Multi Class Classification ,SMOTE+Tomek Links

KATA PENGANTAR

Puji syukur Alhamdulillah senantiasa penulis panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tesis dengan judul **" *Combine* Sampling-Least Square Support Vector Machine Untuk Klasifikasi Multi Class Imbalanced Data"**. Tesis ini disusun sebagai salah satu syarat untuk menyelesaikan studi pada Program Studi Statistika Program Pascasarjana Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Selesainya tesis ini tidak lepas dari bantuan berbagai pihak, untuk itu penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak Suhartono, M.Sc., selaku Ketua Jurusan Statistika ITS yang telah mendukung dan memfasilitasi proses pengerjaan Tesis ini.
2. Ibu Santi Wulan Purnami, M.Si, Ph.D dan Bapak Dr. rer.pol. Dedy Dwi Prastyo, M.Si., selaku dosen pembimbing yang telah banyak meluangkan waktunya untuk memberikan dorongan, petunjuk, bimbingan dan arahan kepada penulis untuk dapat menyelesaikan tesis.
3. Bapak Dr. Purhadi, M.Sc. dan Ibu Santi Puteri Rahayu, M.Si., Ph.D selaku dosen penguji yang telah memberikan masukan dan kritikan yang membangun untuk kebaikan Tesis ini.
4. Bapak dan Ibu dosen pengajar serta staf Jurusan Statistika FMIPA ITS Surabaya, yang dengan tulus ikhlas telah memberikan bekal ilmu selama penulis mengikuti studi.
5. Bapak, Ibu dan Keluarga yang telah memberikan dorongan, semangat dan Doa untuk dapat menyelesaikan Tesis ini
6. Rani Trapsilasiwi dan Mbak Hartayuni Sain yang telah meluangkan waktunya untuk membantu pengerjaan Tesis ini
7. Rizki Fauzi S1 Statistika 2012 yang telah meluangkan waktunya untuk diskusi dalam hal komputasi.
8. Rekan-rekan seperjuangan dari D3 hingga S2 antara lain Fastha, Zubdatu, Millatur, Masnatul, Alkindi, Mbak Mudhiafatur, mbak Fina Mas Adi, Mas

Sukri Adnan Sangadji dan Teman-teman mahasiswa Magister Statistika angkatan 2014 atas segala bantuan, kekompakan dan kebersamaannya selama di ITS Surabaya.

9. Semua pihak yang telah banyak membantu penulis yang tidak dapat penulis sebutkan satu-persatu.

Akhirnya penulis berharap semoga tesis ini dapat memberikan manfaat dan sumbangan untuk menambah wawasan keilmuan bagi pembaca. Penulis menyadari bahwa tulisan ini tentu masih banyak kekurangan. Oleh karena itu, penulis terbuka lebar menerima kritik dan saran.

Surabaya, Februari 2016

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Klasifikasi merupakan salah satu bidang kajian dalam *machine learning*. Analisis Klasifikasi adalah proses menemukan model terbaik dari *classifier* untuk memprediksi kelas dari suatu objek atau data yang label kelasnya tidak diketahui (Han dan Kamber, 2001). Metode klasifikasi dapat dilakukan dengan pendekatan parametrik dan pendekatan nonparametrik. Metode klasifikasi dengan pendekatan parameterik yang sering digunakan antara lain: Analisis Regresi Logistik, Analisis Diskriminan dan Analisis Regresi Probit. Regresi logistik dan regresi probit memiliki kelemahan, yaitu nilai yang dihasilkan model regresi logistik dan probit berupa nilai probabilitas yang dirasa kurang praktis (Yohannes dan Webb, 1999). Pada analisis diskriminan, terikat oleh beberapa asumsi yaitu data berdistribusi normal *multivariate* dan matrik kovarian yang sama untuk setiap populasi (Breiman, Friedman, Olshen dan Stone, 1984).

Selama dekade terakhir ini telah banyak metode *machine learning* yang dikembangkan untuk membantu klasifikasi tanpa terikat oleh asumsi dan memberikan fleksibilitas analisis data yang lebih besar tetapi tetap menghasilkan tingkat akurasi yang tinggi dan mudah dalam penggunaannya, antara lain *Multivariate Adaptive Regression Spline* (MARS), *Feed-Forward Neural Network* (FFNN), *K-Nearest Neighbors* (K-NN), *Classification Adaptive Regression Tree* (CART), *Artificial Neural Network* (ANN), dan *Support Vector Machine* (SVM) (Scholkopf dan Smola, 2002).

Menurut Vapnik (1995), metode *Support Vector Machine* (SVM) merupakan metode *machine learning* yang baru, sangat berguna dan sangat berhasil dalam melakukan prediksi, baik dalam kasus klasifikai maupun regresi. Prinsip dasar SVM adalah *linier classifier* dan selanjutnya dikembangkan untuk masalah *nonlinier* dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi (Cortez dan Vapnik, 1995). Metode SVM menemukan solusi global optimal dan bekerja dengan memetakan data *training* ke ruang berdimensi

tinggi kemudian dalam ruang berdimensi tinggi, akan dicari klasifikasi yang mampu memaksimalkan *margin* antara dua kelas data (Gunn, 1998). Secara sederhana, konsep SVM adalah usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah *class* pada *input space* (Rahman dan Purnami, 2012). Metode *Support Vector Machine* (SVM) dikembangkan berdasarkan pada *statistical learning teory* dan *Structural Risk Minimization* (SRM). Jika dibandingkan dengan metode *machine learning* lainnya, SVM mengimplementasikan prinsip *Structural Risk Minimization* (SRM) daripada *Empirical Risk Minimization* (ERM) (Vapnik,1995). Menurut teori *Structural Risk Minimization* (SRM), SVM telah memperlihatkan performa sebagai metode yang bisa mengatasi masalah *overfitting* dengan cara meminimalkan batas atas pada *generalization error*, yang menjadi alat yang kuat untuk *supervised learning*. SVM dapat menangani sampel besar atau kecil, *nonlinier*, *high dimensional*, *over learning* dan masalah lokal minimum (Guo dkk, 2014).

Penelitian sebelumnya tentang klasifikasi menggunakan SVM yaitu Cheng, Yang dan Liu (2011) menggunakan klasifikasi SVM untuk diagnosis *breast cancer,*hasilnya menunjukan bahwa SVM menghasilkan akurasi yang tinggi. Rahman dan Purnami (2012) membandingkan klasifikasi data dengan regresi logistik ordinal dan SVM, hasilnya menunjukan bahwa SVM memiliki ketepatan klasifikasi yang lebih baik dibandingkan regresi logistik ordinal. Novianti dan Purnami (2012) membandingkan klasifikasi SVM dengan regresi logistik, hasil menunjukan bahwa akurasi klasifikasi dengan SVM lebih baik daripada regresi logistik. Haerdle, Prastyo dan Hafner (2014) melakukan prediksi kegagalan peminjaman kredit bank dengan membandingkan antara SVM, analisis diskriminan, probit dan logit, hasil menunjukan bahwa klasifikasi SVM lebih baik dibandingan dengan metode klasifikasi yang lain. Lee dan To (2010) membandingkan SVM dan *Back Propagation Neural Network* untuk mengevaluasi krisis keuangan perusahaan, hasil menunjukan bahwa SVM menghasilkan hasil yang lebih baik dibandingkan dengan *Back Propagation Neural Network.* Priya dan Aruna (2012), menggunakan SVM dan *Probabilistic Neural Network* untuk diagnosis diabetes, hasil menunjukan bahwa SVM

menghasilkan akurasi (97,68%) yang lebih tinggi daripada *Probabilistic Neural Network* (89,60%).

Pada algoritma SVM, terdapat *quadratic programming* yang merupakan suatu kompleksitas komputasi dari algoritma SVM yang biasanya intensif digunakan, karena dengan *quadratic programming* dapat diperoleh solusi optimal dalam menentukan fungsi *lagrange, dari* fungsi *langrange* akan digunakan dalam perhitungan nilai parameter bobot dan bias. *Quadratic programming* tidak efisien apabila diterapkan pada dimensi ruang yang lebih tinggi karena komputasi akan semakin kompleks dan akan sangat lama. Oleh karena itu, dikembangkan metode *Least Square Support Vector Machine* (LS-SVM).

Suykens dan Vandewalle (1999), memperkenalkan modifikasi SVM yang diberi nama *Least Square Support Vector Machine* (LS-SVM). LS-SVM lebih baik dibandingkan dengan SVM standart dalam hal proses perhitungan, konvergensi cepat dan presisi yang tinggi. Jika SVM menggunakan fungsi *constrain* yang hanya berupa pertidaksamaan maka LS-SVM diformulasikan menggunakan fungsi constrain berupa persamaan. Sehingga solusi LSSVM dihasilkan dengan menyelesaikan persamaan linier. Hal ini berbeda dengan SVM, dimana solusinya dihasilkan melalui penyelesaian *quadratic programming* (Suyken dan Vandewalle, 1999a,1999b,1999c; Bhavsar dkk, 2012; Trapsilasiwi, 2013). Abdullah (2013) melakukan penelitian menggunakan *Least Square Support Vector Machine (*LS-SVM) dalam aplikasi *Wide Area Control System* (WACS) untuk meredam osilasi pada sistem tenaga kerja listrik dua area. Hasil menunujukan bahwa Aplikasi WACS berbasis LS-SVM- dapat memperbaiki nilai *eigenvalue* lokal dan inter-area pada sistem tenaga listrik lebih baik daripada WACS berbasis SVM.

LS-SVM berisi beberapa parameter yang mempengaruhi performansi sehingga diperlukan sebuah algoritma untuk pemilihan optimasi parameter. Optimasi parameter untuk klasifikasi banyak macamnya, antara lain menggunakan *Genetic Algoritm* (GA) (Haerdle, Prastyo dan Hafner, 2014), *Particle Swarm Optimization (PSO)* (Kennedy dan Eberhart, 1995), *Gravitational Search Algoritm* (Rashedi,2009), *Grid Search* (Chen, Lin dan Scholkopf, 2011), *Particle Swarm Optimization and Gravitational Search Algoritm* (PSO-GSA) (Mirjalili,

2010;Trapsilasiwi, 2013) dan lain-lain. Menurut Mirjalili (2010), keunggulan dalam pemilihan PSO-GSA untuk optimisasi parameter kernel dan nilai C yaitu kesederhanaan implementasi dan kecepatan menuju konvergen pada suatu solusi yang cukup baik.

Pada kehidupan nyata, khususnya di bidang medis seringkali ditemui klasifikasi dalam kasus *multi class*. Klasifikasi *multi class* adalah mengklasifikasikan setiap titik data pada kelas yang berbeda, dimana kelas lebih dari dua. Klasifikasi SVM dan LS-SVM yang semula untuk *binary class* akan dimodifikasi dengan menggunakan pendekatan *multi class.* Ada beberapa pendekatan yang sering digunakan untuk kasus *multi class* antara lain *One Againt All* (OAA), *One Againt One* (OAO) dan *Directed Acyclic Graph* (DAG) (Hsu dan Lin, 2002). Zheng dkk (2011) menerapkan metode LS-SVM *multi class* untuk diagnosis *power transformer*, hasilnya menyatakan bahwa metode LS-SVM menggunakan pendekatan OAO lebih baik dibandingkan dengan pendekatan *multi class* lainnya. Trapsilasiwi (2013) menyatakan bahwa klasifikasi *multi class* dengan pendekatan *One Againt One* (OAO) lebih baik dibandingkan dengan menggunakan pendekatan *One Againt All* (OAA).

Ada dua kondisi himpunan data yaitu *balanced* dan *imbalanced* data. Pada klasifikasi *multi class* seringkali ditemui kondisi himpunan data *imbalanced*. *Imbalanced data* merupakan kondisi data yang tidak berimbang antara kelas data satu dengan kelas data yang lain. Kelas data yang banyak merupakan kelas mayoritas atau kelas negatif sedangkan kelas data yang sedikit merupakan kelas minoritas atau kelas positif. Kondisi *imbalanced* data menjadi masalah dalam klasifikasi karena mesin *classifier learning* akan condong memprediksi ke kelas data yang banyak (mayoritas) dibanding dengan kelas minoritas. Akibatnya, dihasilkan akurasi prediksi yang baik terhadap kelas data *training* yang banyak (kelas mayoritas) sedangkan untuk kelas data *training* yang sedikit (kelas minoritas) akan dihasilkan akurasi prediksi yang buruk (Japkowicz dan Stephen, 2002 ; Chawla, 2003; Sain, 2013).

Masalah *imbalanced* data telah menjadi perhatian utama karena kinerja algoritma secara signifikan akan menurun. Masalah *imbalanced data* terjadi diberbagai macam konteks seperti informasi pengambilan dan penyaringan (Lewis

dan Catlett, 1994), klasifikasi teks (Chawla dkk, 2002), deteksi penipuan kartu kredit (Wu dan Chang, 2003), deteksi tumpahan minyak dari pencitraan satelit (Kubat dkk, 1998), diagnosais medis (Kononenko, 2001; Sain, 2013 ; Trapsilasiwi, 2013) dan lain-lain.

Menurut Choi (2010), ada tiga pendekatan metode *learning* untuk mengatasi masalah *imbalanced data*. Pendekatan pertama yaitu menggunakan level data (*Sampling-Based Approach)*. Pendekatan kedua yaitu pada level algoritma. Pendekatan ketiga yaitu dengan metode *ensemble learning.* Pendekatan *sampling* pada data yang *imbalanced* menyebabkan tingkat *imbalanced data* semakin kecil dan klasifikasi dapat dilakukan dengan tepat (Solberg dan Solberg, 1996). *Sampling based approach* yaitu memodifikasi distribusi data training sehingga kedua kelas data (negatif maupun positif) dipresentasikan dengan baik di dalam data training. Pendekatan *Sampling* dibedakan menjadi dua yaitu *oversampling* dan *undersampling.*

Metode *oversampling* dilakukan untuk menyeimbangkan jumlah distribusi data dengan cara meningkatkan jumlah data kelas minor. Masalah umum yang akan muncul dari metode *oversampling* adalah masalah *overfitting*, yang menyebabkan aturan klasifikasi menjadi semakin spesifik meskipun akurasi untuk data *training* semakin membaik. Metode *undersampling* dilakukan dengan cara mengurangi jumlah data kelas mayor agar data menjadi seimbang. Metode ini akan kehilangan informasi dari data karena banyak data yang dihilangkan, yang mengandung banyak informasi sehingga efektivitas klasifikasi menurun sedangkan penghapusan data yang tidak relevan, berlebihan ataupun noise dapat mengakibatkan efektifitas klasifikasi meningkat (He dan Garcia, 2009; Chawla, 2002; Sain, 2013).

Salah satu metode *oversampling* adalah dengan menggunakan *Synthetic Minority Oversampling Technique* (SMOTE), yang pertama kali diperkenalkan oleh Chawla (2002). Pendekatan ini bekerja dengan membuat "*synthetic*" data, yaitu data replikasi dari data minor. Algoritma SMOTE digunakan oleh Chawla (2002) pada klasifikasi *imbalanced* data dengan *decision tree*. Menurut Chawla (2002), metode *Synthetic Minority Oversampling Technique* (SMOTE) merupakan metode yang kuat untuk menangani masalah *imbalanced* data dan

telah sukses dalam berbagai macam kasus aplikasi akan tetapi masalah umum yang akan muncul dari metode *oversampling* adalah masalah *overfitting*. Trapsilasiwi (2013), menerapkan metode SMOTE *Least Square SVM* (LS-SVM*)* PSO-GSA pada data medis untuk menangani masalah *imbalanced data multi class* dengan menggunakan partisi 70-30% dan 5-*fold cross validation.* Hasil menunjukan bahwa metode SMOTE LS-SVM PSO-GSA lebih baik dibandingkan dengan metode LS-SVM tanpa adanya penambahan SMOTE dan PSO GSA. Akan tetapi, metode SMOTE LS-SVM ini belum memberikan hasil yang memuaskan pada kedua data percobaan. Pada hasil terlihat kalau masih terjadi *missclassification* yang cukup besar dan *overfitting*. Akurasi tertinggi SMOTE LS-SVM PSO-GSA pada data kanker serviks sekitar 59,4%. jauh lebih rendah daripada akurasi data kanker payudara sekitar 96,9%. Akurasi pada *training* jauh lebih tinggi daripada akurasi pada *testing* atau terjadi *overfitting.* Oleh karena itu, dilakukan perbaikan klasifikasi pada tahap *preprocessing* menangani *imbalanced* data.

Estabrooks dkk (2004) menyatakan bahwa metode penggabungan *undersampling* dan *oversampling* merupakan metode yang sangat efektif untuk menangani masalah *imbalanced* data. Salah satu metode *undersampling* adalah *Tomek Links* (Tomek, 1997). Cara kerja Tomek Links yaitu dengan menghapus data kelas negatif (mayor) yang merupakan kasus *borderline* atau *noise*. Batista dkk (2003) menggunakan metode gabungan *undersampling* dan *oversampling* (SMOTE+Tomek Links) pada klasifikasi masalah pengkajian protein dalam bioinformatika dengan *decision tree*. Penggunaan metode SMOTE+Tomek Links merepsentasikan hasil yang sangat baik untuk masalah *imbalanced* data. Gaudio dkk (2011) melakukan perbandingan metode *imbalanced* data pada ekstraksi *setting* multi bahasa. Menurut Gaudio dkk (2011) penggunaan metode SMOTE+Tomek Links efektif untuk menangani masalah *higher imbalanced* data. Sain (2013), menerapkan metode *Combine Sampling* (SMOTE+Timek Link) dengan metode SVM 5-*fold cross validation*, yang diterapkan pada data medis. Hasil dari Sain (2013) menunjukan bahwa dengan metode *combine sampling* (SMOTE+Tomek Links) secara umum lebih baik dari metode SMOTE dan Tomek Links. Akan tetapi, penerapan metode *Combine Sampling*

(SMOTE+Tomek Links) penelitian sebelumnya masih digunakan untuk klasifikasi *binary class*.

Berdasarkan kelebihan dan kekurangan metode yang telah disebutkan sebelumnya, maka peneliti mengusulkan metode penggabungan SMOTE dan Tomek Links *Least Square Support Vector Machine* (LS-SVM) untuk klasifikasi *multi class imbalanced* data. Metode tersebut akan diterapkan pada data medis (Kanker payudara, kanker serviks dan *Thyroid*). Penggabungan metode SMOTE dan Tomek Links ini dinamakan metode *Combine Sampling*. Metode *Combine Sampling* (SMOTE+Tomek Links) digunakan untuk *preprocessing* menangani *imbalance* data. LS-SVM *One Againt One* (OAO) digunakan untuk klasifikasi *multi class*. Optimasi parameter fungsi kernel RBF σ dan C menggunakan PSO-GSA. Penelitian ini mengadopsi dari penelitian sebelumnya (Trapsilasiwi, 2013) dan (Sain, 2013; Batista dkk, 2003; Batista dkk 2004; Gaudio,2013).

1.2 Rumusan Masalah

Kondisi *imbalanced* data menjadi masalah dalam klasifikasi *multi class* karena mesin *classifier learning* akan condong memprediksi ke kelas data yang banyak (mayoritas) dibanding dengan kelas minoritas. Akibatnya, dihasilkan akurasi prediksi yang baik terhadap kelas data *training* yang banyak (kelas mayoritas) sedangkan untuk kelas data *training* yang sedikit (kelas minoritas) akan dihasilkan akurasi prediksi yang buruk. Dari uraian diatas maka permasalahan dalam penelitian ini adalah bagaimana algoritma *Combine Sampling* (*SMOTE+Tomek Links)* pada klasifikasi *multi class imbalanced* data dan penerapan *Least Square Support Vector Machine* (LS-SVM) menggunakan *Combine Sampling* (SMOTE+Tomek Links) sebagai *preprocessing* mengatasi *imbalanced* pada data medis.

1.3 TujuanPenelitian

Tujuan penelitian ini adalah sebagai berikut :

1. Mendesain algoritma *Combine Sampling (SMOTE+Tomek Links)* untuk kasus klasifikasi *multi class imbalanced* data.
2. Menerapkan metode *Combine Sampling (SMOTE+Tomek Links)* LS-SVM untuk klasifikasi *multi class imbalanced* dengan menggunakan data medis

1.4 Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah

1. Memberikan metode alternatif untuk klasifikasi *multi class imbalanced* data dengan menggunakan metode *Combine Sampling Least Square* SVM (LS-SVM).
2. Memberikan informasi mengenai pengklasifikasian *multi class imbalanced* data menggunakan metode *Combine Sampling Least Square* SVM (LS-SVM) pada data medis.
3. Menambah keilmuan Statistika dibidang klasifikasi data mining *machine learning.*

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut.

1. Fungsi kernel yang digunakan untuk klasifikasi adalah fungsi kernel Gaussian Radial Basis (RBF)
2. Klasifikasi *multi class* dengan menggunakan pendekatan algoritma *One Againt One* (OAO).
3. Studi kasus yang digunakan hanya menggunakan data medis *imbalanced data multi class.*

BAB II
TINJAUAN PUSTAKA

Pada bab ini membahas tentang metode-metode yang digunakan yaitu Deteksi Missing value dan Outlier, SMOTE (*Synthetic Minority Oversampling Technique*), Tomek Links, LS-SVM (*Least Square SVM*), OAO (*One Against One*), PSO-GSA (*Particle Swarm Optimization-Gravitational Search Algorithm*), evaluasi performansi metode klasifikasi, Q-*fold crossvalidation*, Uji Friedman dan penelitian-penelitian yang telah dilakukan mengenai klasifikasi pada kasus *imbalanced data*.

2.1 Preprocesssing Data

Metode yang digunakan untuk preprocessing data adalah sebagai berikut.

2.1.1 Deteksi Missing Value

Missing value atau *missing data* merupakan gangguan yang biasa ditemukan peneliti dalam data yang akan dianalisis. *Missing value* dapat muncul pada saat pengumpulan data, entri data, maupun pada saat *interviewer* mengisikan jawaban responden pada lembar kuesioner. Jika persentase data *missing value* melebihi 30%, maka data boleh dihapus sedangkan jika persentase data *missing value* kurang 30%, maka data missing diimputasi dengan nilai mean jika data kuantitatif dan modus jika data kualitatif (Hair, 1995).

2.1.2 Deteksi Outlier

Data *outlier* merupakan observasi yang nilainya sangat menyimpang dibandingkan hasil pengamatan lainnya. Jika terdapat oulier maka data ke-*i* dibuang. Pada dataset *univariat*e, pendeteksian data *outlier* dapat dilakukan dengan melalui *boxplot* atau diagram *steam-and-leaf*. Pengujian *outlier secara multivariate* adalah (Morrison, 2005):

H_0 : Data ke-*i* tidak outlier

H_1 : Data ke-*i* outlier

Statistik Uji : $$\boldsymbol{F}_i = \frac{(n-p-1)n\ \mathbf{D}_i^2}{p(n-1)^2\ n\ p\ \mathbf{D}_i^2} \qquad (2.1)$$

dimana $\mathbf{D}_i^2 = (\mathbf{x}_i - \bar{\mathrm{x}})^T \boldsymbol{S}^{-1} (\mathbf{x}_i - \bar{\mathrm{x}})$

D_i^2 adalah jarak mehalanobis, $\bar{\mathrm{x}}$ adalah rata-rata dari pengamatan, x_i adalah data ke-*i*, n adalah banyak sampel dan *p* adalah banyak prediktor dan S adalah matrik varians covarians.

Daerah kritis : Tolak H_0 jika $F_{hitung\ i} > F_{\alpha;p,n-p-1}$ atau p-Value<α

2.2 *Preprocesssing Imbalanced* Data

Metode yang digunakan untuk preprocessing imbalanced data adalah sebagai berikut:

2.2.1 Syntetics Minority Oversampling Technique (SMOTE)

Synthetic Minority Oversampling Technique (SMOTE) merupakan salah satu metode *oversampling* yaitu teknik pengambilan sampel untuk meningkatkan jumlah data pada kelas minoritas dengan cara mereplikasi jumlah data pada kelas minoritas secara acak sehingga jumlahnya sama dengan data pada kelas mayoritas. Algoritma SMOTE pertama kali diperkenalkan oleh Chawla (2002). Pendekatan ini bekerja dengan membangkitkan "synthetics" data yaitu data baru yang direplikasi dari data minor. Algoritma SMOTE bekerja dengan mencari k-*nearest neighbour*, yaitu mengelompokan data berdasarkan tetangga terdekat. Tetangga terdekat dipilih berdasarkan jarak *euclidean* antara sepasang data. Teknik ini hampir mirip dengan *clustering*.

Misalkan diberikan data dengan *p* variabel yaitu $\mathbf{x}^T = \left[\mathrm{x}_1, \mathrm{x}_2, ..., \mathrm{x}_p\right]$ dan $\mathbf{z}^T = \left[z_1, z_2, ..., z_p\right]$ maka jarak *Eulidean* $\mathrm{d}(\mathbf{x}, \mathbf{z})$ secara umum pada persamaan (2.1)

$$\mathrm{d}(\mathbf{x}, \mathbf{z}) = \sqrt{\left(\mathrm{x}_1 - \mathrm{z}_1\right)^2 + \left(\mathrm{x}_2 - \mathrm{z}_2\right)^2 + ... + \left(\mathrm{x}_\mathrm{p} - \mathrm{z}_\mathrm{p}\right)^2} \qquad (2.2)$$

Jika *p*=2 maka diperoleh jarak Eulidean $\mathrm{d}(\mathbf{x}, \mathbf{z})$ yaitu

$\mathrm{d}(\mathbf{x}, \mathbf{z}) = \sqrt{\left(\mathrm{x}_1 - \mathrm{z}_1\right)^2 + \left(\mathrm{x}_2 - \mathrm{z}_2\right)^2}$.

Pembangkitan data *Synthetics* dilakukan dengan menggunakan persamaan (2.3) :

$$\mathrm{x}_{syn} = \mathrm{x}_i + \left(\mathrm{x}_{knn} - \mathrm{x}_i\right) \times \gamma \qquad (2.3)$$

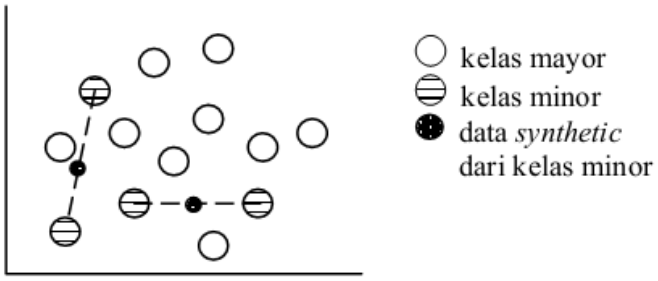
dimana x_{syn} adalah data hasil replikasi.

x_i adalah data ke-i dari kelas minor.

x_{knn} adalah data dari kelas minor yang memiliki jarak terdekat dari x_i

γ adalah bilangan random antara 0 dan 1 (Choi, 2010).

Ilustrasi tentang algoritma SMOTE dapat dilihat pada Gambar 2.1.



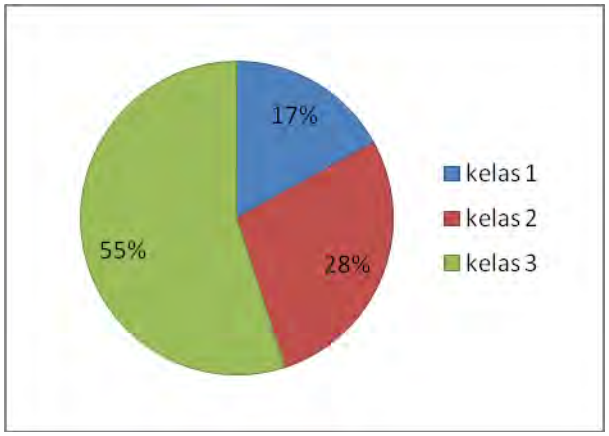
Gambar 2.1 Ilustrasi Algoritma SMOTE (Diperoleh dari Trapsilasiwi, 2013)

Sebagai contoh ilustrasi algoritma SMOTE diatas maka diberikan sebuah contoh simulasi pada Tabel 2.1.

Tabel 2.1 Data Simulasi

Data ke-	X_1	X_2	Y	Data ke-	X_1	X_2	Y
1	1	1	1	10	7	3	3
2	2	3	1	11	7	4	3
3	3	4	1	12	5	5	3
4	3	2	2	13	6	5	3
5	3	3	2	14	8	2	3
6	4	1	2	15	8	4	3
7	5	4	2	16	9	2	3
8	6	2	2	17	9	5	3
9	7	1	3	18	10	3	3

Contoh simulasi data pada Tabel 2.1, yang merupakan kasus *imbalanced* data, dapat ditunjukkan dengan Gambar 2.2.



Gambar 2.2 Persentase Masing-masing kelas

Tahapan yang dilakukan pada algoritma SMOTE adalah:

1. Setiap data pada kelas minor akan direplikasi mencari tetangga terdekat (x_{knn}) dengan menggunakan jarak eulidean. Pembangkitan data sintetis dari minoritas (Y=1) adalah sebagai berikut:

Data ke-1 dan data ke-2 :

$$d\left(\begin{bmatrix}1\\1\end{bmatrix},\begin{bmatrix}2\\3\end{bmatrix}\right)=\sqrt{(1-2)^2+(1-3)^2}=\sqrt{5}$$

Data ke-2 dan data ke-3 :

$$d\left(\begin{bmatrix}2\\3\end{bmatrix},\begin{bmatrix}3\\4\end{bmatrix}\right)=\sqrt{(2-3)^2+(3-4)^2}=\sqrt{2}$$

Data ke-1 dan data ke-3 :

$$d\left(\begin{bmatrix}1\\1\end{bmatrix},\begin{bmatrix}3\\4\end{bmatrix}\right)=\sqrt{(1-3)^2+(1-4)^2}=\sqrt{13}$$

Pembangkitan data sintetis dari minoritas (Y=2) adalah sebagai berikut:

Data ke-4 dan data ke-5 :

$$d\left(\begin{bmatrix}3\\2\end{bmatrix},\begin{bmatrix}3\\3\end{bmatrix}\right)=\sqrt{(3-3)^2+(2-3)^2}=\sqrt{1}$$

Data ke-5 dan data ke-6 :

$$d\left(\begin{bmatrix}3\\3\end{bmatrix},\begin{bmatrix}4\\1\end{bmatrix}\right)=\sqrt{(3-4)^2+(3-1)^2}=\sqrt{5}$$

Dari perhitungan diatas maka dapat diambil dua jarak *Euclidean* yang terdekat pada kelas 1(Y=1) yaitu $\sqrt{2}$ dan $\sqrt{5}$ maka kelas 1 akan direplikasi dua kali. Jumlah data kelas 1 yang semula berjumlah 3 maka setelah direplikasi sebanyak 2 kali akan menjadi 9 data. Pada kelas 2 (Y=2) diambil satu jarak Euclidean yang terdekat yaitu $\sqrt{1}$ maka kelas 2 (Y=2) akan direplikasi 1 kali. Jumlah data kelas 2 yang semula berjumlah 5 maka setelah direplikasi sebanyak 1 kali akan menjadi

10 data. Hasil distribusi data simulasi SMOTE, dapat dilihat pada Tabel 2.2 dan data simulasi setelah SMOTE, dapat dilihat pada Tabel 2.3.

2. Menghitung *Synthetic data* dengan menggunakan rumus :

$$x_{syn} = x_i + (x_{knn} - x_i) \times \gamma$$

Perhitungan data sintesis (data hasil replikasi) pada kelas 1 adalah

$$x_{syn} = [1,1] + ([2,3] - [1,1]) \times 0,3 = [1,3;1,6]$$

$$x_{syn} = [2,3] + ([3,4] - [2,3]) \times 0,3 = [2,3;3,3]$$

$$x_{syn} = [3,4] + ([2,3] - [3,4]) \times 0,3 = [2,7;3,7]$$

Tabel 2.2 Distribusi Data Simulasi Sebelum dan Setelah SMOTE

Kelas mayor	Kelas minor	Replikasi	Kelas mayor	Kelas minor baru
10 (55%)	3 (17%)	2	10 (34%)	9 (32%)
	5 (28%)	1		10(34%)

Tabel 2.3 Data Simulasi Setelah Menggunakan SMOTE

Data ke-	X_1	X_2	Y	Data ke-	X_1	X_2	Y	Data ke-	X_1	X_2	Y
1	1	1	1	11	7	4	3	21*	2	3	1
2	2	3	1	12	5	5	3	22*	2,3	3,3	1
3	3	4	1	13	6	5	3	23*	3	4	1
4	3	2	2	14	8	2	3	24*	2,7	3,7	1
5	3	3	2	15	8	4	3	25*	3	2	2
6	4	1	2	16	9	2	3	26*	3	3	2
7	5	4	2	17	9	5	3	27*	4	1	2
8	6	2	2	18	10	3	3	28*	5	4	2
9	7	1	3	19*	1	1	1	29*	6	2	2
10	7	3	3	20*	1,3	1,6	1				

*) Synthetic Data

2.2.2 Tomek Links

Tomek Links merupakan salah satu metode *undersampling*, yang diperkenalkan oleh Tomek pada Tahun 1997. Metode ini bekerja dengan menghapus data kelas negatif (mayoritas) yang merupakan kasus *borderline* atau yang memiliki kesamaan karakteristik. Tomek Links dapat digunakan sebagai

metode pembersihan data dari *noise*. Untuk setiap data, jika satu tetangga yang paling dekat memiliki kelas label yang berbeda dengan data tersebut maka data mayor akan dihapus karena dianggap sebagai *noise* atau *misclassification*. Diberikan dua sampel **x** dan **z** milik kelas yang berbeda, dan d(**x**,**z**) adalah jarak antara **x** dan **z**. Sepasang (x,z) disebut Tomek Links jika tidak ada sampel **z***, sehingga $d(\mathbf{x},\mathbf{z}^*) < d(\mathbf{x},\mathbf{z})$ atau $d(\mathbf{z},\mathbf{z}^*) < d(\mathbf{z},\mathbf{x})$ (Batista, Bazzan dan Monard, 2003). Jika dua sampel membentuk *Tomek Links*, maka salah satu dari kedua sampel adalah data *noise* atau kedua contoh adalah *borderline.* Misal diberikan data, seperti pada Tabel 2.1, dimana (Y=3) merupakan sampel dari kelas mayoritas dan (Y=1) dan (Y=2) kelas minoritas, sehingga contoh hasil penerapannya adalah sebagai berikut. Dengan menggunakan rumus jarak eulidean yaitu:

$$d(\mathbf{x},\mathbf{z}) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2} \tag{2.4}$$

Data dari kelas mayor (Y=3) dan data dari kelas minor (Y=1) dihitung jarak eulidean didapatkan jarak eulidean yang terdekat adalah y_{12}=(5,5) dengan y_3=(3,4) (dapat dilihat pada Gambar 2.3), akan tetapi y_{12}=(5,5) dan y_3=(3,4) bukan teridentifikasi kasus tomek Links.

$$d(y_{12}, y_7) = \sqrt{(5-5)^2 + (5-4)^2} = \sqrt{1}$$
$$d(y_3, y_7) = \sqrt{(3-5)^2 + (4-4)^2} = \sqrt{4}$$

Kesimpulan yang diperoleh adalah $d(y_{12}, y_7) = \sqrt{1} < d(y_{12}, y_3) = \sqrt{5}$ atau $d(y_3, y_7) = \sqrt{4} < d(y_{12}, y_3) = \sqrt{1}$, dengan demikian kedua titik y_9 dan y_1 merupakan bukan kasus Tomek Links karena memenuhi syarat dari definisi kasus *Tomek Links*.

Kemudian berpindah kelas mayor (Y=3) dengan kelas minor (Y=2). Titik lain yang terdeteksi (**z***) berada dekat dengan kedua kelas dapat berasal dari kelas mayor ataupun minor.

i. Diambil data dari kelas mayor (Y=3) dan data dari kelas minor (Y=2). Misal data dari kelas mayor $y_{10} = (7,3)$ dan data dari kelas minor $y_8 = (6,2)$

kemudian dicari jarak eulidean : $d(y_{10},y_8)=\sqrt{(7-6)^2+(3-2)^2}=\sqrt{2}$

kemudian mencari titik lain yang terdeteksi (**z***) berada dekat dengan antara kedua titik y_{10} dan y_8 adalah $y_{14}=(8,2)$, sehingga

$$d(y_{10},y_{14})=\sqrt{(7-8)^2+(3-2)^2}=\sqrt{2}$$
$$d(y_8,y_{14})=\sqrt{(6-8)^2+(2-2)^2}=\sqrt{4}$$

Kesimpulan yang diperoleh adalah $d(y_{10},y_{14})=\sqrt{2}\geq d(y_{10},y_8)=\sqrt{2}$ atau $d(y_8,y_{14})=\sqrt{4}>d(y_{10},y_8)=\sqrt{2}$, dengan demikian kedua titik y_{10} dan y_8 merupakan kasus Tomek Links karena tidak memenuhi syarat dari definisi kasus Tomek Links sehingga titik dari kelas mayor $y_{10}=(7,3)$ akan dihapus.

Dengan proses cara yang sama pada dari kelas mayor $y_9=(7,1)$ dan data dari kelas minor $y_8=(6,2)$ dengan titik terdekat (**z***) $y_{14}=(8,2)$ diperoleh keputusan kedua titik y_9 dan y_8 merupakan kasus Tomek Links sehingga titik dari kelas mayor $y_9=(7,1)$ akan dihapus.

ii. Diambil data dari kelas mayor (Y=3) dan data dari kelas minor (Y=2). Misal data dari kelas mayor $y_{12}=(5,5)$ dan data dari kelas minor $y_7=(5,4)$

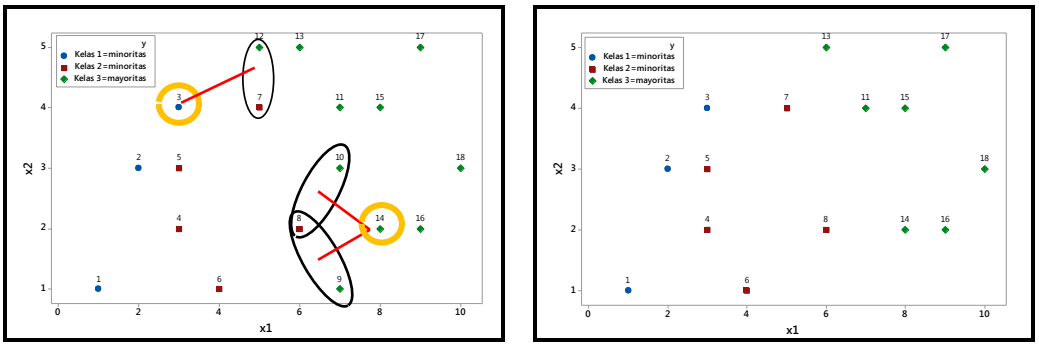
kemudian dicari jarak eulidean : $d(y_{12},y_7)=\sqrt{(5-5)^2+(5-4)^2}=\sqrt{1}$

selanjutnya mencari titik lain yang terdeteksi (**z***) berada dekat dengan antara kedua titik y_{12} dan y_7 adalah $y_3=(3,4)$, sehingga

$$d(y_{12},y_3)=\sqrt{(5-3)^2+(5-4)^2}=\sqrt{5}$$
$$d(y_7,y_3)=\sqrt{(5-3)^2+(4-4)^2}=\sqrt{4}$$

Kesimpulan yang diperoleh adalah $d(y_{12},y_3)=\sqrt{5}>d(y_{12},y_7)=\sqrt{1}$ atau $d(y_7,y_3)=\sqrt{4}>d(y_{12},y_7)=\sqrt{1}$, dengan demikian kedua titik y_{12} dan y_7 merupakan kasus Tomek Links karena tidak memenuhi syarat dari definisi kasus Tomek Links sehingga titik $y_{12}=(5,5)$ akan dihapus.

Ilustrasi metode Tomek Link untuk menangani *imbalanced data*, seperti pada Gambar 2.3.

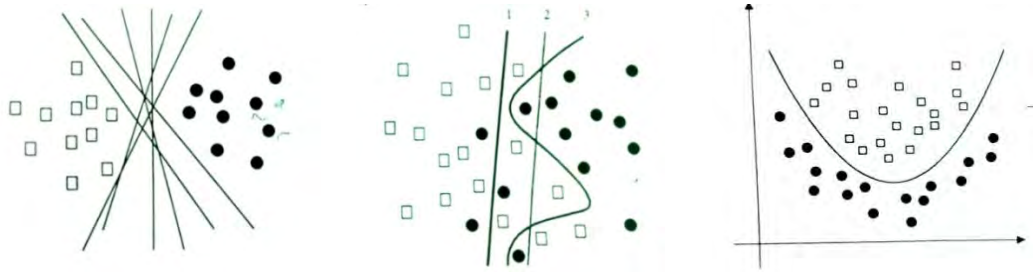


(a) (b)

Gambar 2.3 Ilustrasi Tomek Links: Plot data awal (a) dan Plot data hasil metode Tomek Links (b)

2.3 Support Vector Machine (SVM)

Support vector machines (SVM) adalah metode pembelajaran *supervised*, diperkenalkan pertama kali oleh Vapnik pada tahun 1995 dan sangat berhasil dalam melakukan prediksi, baik dalam kasus regresi maupun klasifikasi. SVM didasarkan pada prinsip minimalisasi resiko struktural/*structural risk minimization* (SRM). Prinsip induksi ini berbeda dari prinsip minimalisasi resiko empirik yang hanya meminimalkan kesalahan pada proses pelatihan. Pada SVM, fungsi tujuan dirumuskan sebagai masalah optimisasi konveks berbasis *quadratic programming*, untuk menyelesaikan *dual problem*. Menurut Tan, Steinbach dan Kumar (2006), *Support Vector Machine (SVM)* adalah metode klasifikasi yang bekerja dengan cara mencari *hyperplane* dengan margin optimum. *Hyperplane* adalah garis batas pemisah data antar kelas. *Margin* (*m*) adalah jarak antara *hyperplane* dengan data terdekat pada masing-masing kelas. Bidang pembatas pertama membatasi kelas pertama dan bidang pembatas kedua membatasi kelas kedua sedangkan data yang berada pada bidang pembatas merupakan vektor-vektor yang terdekat dengan *hyperplane* terbaik disebut dengan *Support Vector*. SVM untuk klasifikasi dapat bekerja pada kasus klasifikasi linier maupun *nonlinier,* seperti diilustrasikan pada Gambar 2.4.Pada klasifikasi linier, SVM dapat dibedakan menjadi dua yaitu *linierly separable* dan *linierly nonseparable.*



(a) (b) (c)

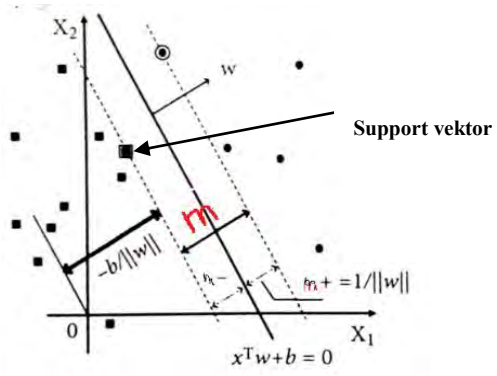
Gambar 2.4. Klasifikasi SVM **:** (a) Klasifikasi *Linier Separable* ; (b) *Linier Nonseparable*; (c)Nonlinier (Diperoleh dari Haerdle, 2014)

2.3.1 SVM *Linierly Separable*

Haerdle, Prastyo dan Hafner (2014) menyatakan, setiap observasi ke-*i* berisi sepasang p prediktor $\mathbf{x}_i^{\mathrm{T}}=(\mathrm{x}_{1i},\mathrm{x}_{2i}...,\mathrm{x}_{pi}),\quad i=1,2,...,n$ dan berpasangan dengan $\mathrm{y}_i\in\{-1,1\}$ maka data dapat dinyatakan dalam himpunan berikut :

$$D_n=\{(\mathbf{x}_1,y_1),...,(\mathbf{x}_n,y_n)\}\in \mathrm{X}\times\{-1,1\}.$$

Jika $\mathbf{x}_i$ adalah anggota kelas (+1) maka $\mathbf{x}_i$ diberi label (target) $y_i=+1$ dan jika tidak maka diberi label (target) $y_i=-1$ sehingga data yang diberikan berupa pasangan $(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),...,(\mathbf{x}_n,y_n)$ merupakan himpunan data *training* dari dua kelas yang akan diklasifikasikan dengan SVM (Gunn, 1998). Pada Gambar 2.5, dapat dilihat bahwa berbagai alternatif bidang pemisah yang dapat memisahkan semua dataset sesuai dengan kelasnya namun bidang pemisah terbaik tidak hanya dapat memisahkan data tetapi juga memiliki margin paling besar (Burges, 1998).



Gambar 2.5 Bidang pemisah terbaik dengan margin (*m*) terbesar (Diperoleh dari Haerdle, 2014)

Fungsi klasifikasi $\mathbf{x}^T\mathbf{w}+b$ berada dalam sebuah keluarga fungsi klasifikasi $\mathcal{F}$ yang terbentuk, yaitu

$$\mathcal{F} = \mathbf{x}^T\mathbf{w} + b\ , \mathbf{w}\epsilon\mathbb{R}^p\ , b \in \mathbb{R}$$

Bidang pemisah (*separating hyperplane*) :

$$f(x) = \mathbf{x}^T\mathbf{w} + b = 0 \tag{2.5}$$

yang membagi ruang *(space)* menjadi dua daerah. Bentuk pada $f(x)$ adalah sebuah garis dalam dua dimensi, sebuah bidang pada tiga dimensi, dan secara umum berupa *hyperplane* pada dimensi yang lebih tinggi. *Hyperplane* dikatakan linier jika merupakan fungsi linier dalam input $\mathbf{x}_i$. Data yang berada pada *margin (m*) disebut dengan *support vector.*

Fungsi pemisah untuk kedua kelas adalah sebagai berikut:

$$\mathbf{x}_i^{\mathrm{T}}\mathbf{w}+\mathrm{b} \geq 1 \text{ untuk } y_i = +1\ ,$$

$$\mathbf{x}_i^{\mathrm{T}}\mathbf{w}+\mathrm{b} \leq -1 \text{ untuk } y_i = -1\ , \tag{2.6}$$

dimana $\mathbf{w}$ adalah vektor bobot (*weight vector*) yang berukuran (*px1*), b adalah posisi bidang relatif terhadap pusat koordinat atau lebih dikenal dengan bias yang bernilai skalar.

$$\mathbf{X} = \begin{bmatrix} \mathrm{x}_{11} & \mathrm{x}_{12} & \cdots & \mathrm{x}_{1n} \\ \mathrm{x}_{21} & \mathrm{x}_{22} & \cdots & \mathrm{x}_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{x}_{n1} & \mathrm{x}_{n2} & \cdots & \mathrm{x}_{np} \end{bmatrix}_{n\times p} \quad \mathbf{x}_i^{\,T} = \begin{bmatrix} \mathrm{x}_{i1} & \mathrm{x}_{i2} & \cdots & \mathrm{x}_{ip} \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} \mathrm{w}_1 \\ \mathrm{w}_2 \\ \vdots \\ \mathrm{w}_p \end{bmatrix} \quad y_i = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Pada Gambar 2.5 menunjukan $\frac{|b|}{\|\mathbf{w}\|}$ adalah jarak bidang pemisah yang tegak lurus dari titik pusat koordinat dan $\|\mathbf{w}\|$ adalah jarak Eulidean (*norm Eulidean*) dari $\mathbf{w}$ .Panjang vektor $\mathbf{w}$ adalah *norm* $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}} = \sqrt{\mathrm{w}_1{}^2 + \mathrm{w}_2{}^2 + \ldots + \mathrm{w}_p{}^2}$. Bidang batas pertama membatasi kelas (+1) sedangkan bidang pembatas kedua membatasi kelas (-1).

Bidang pembatas pertama $\mathbf{x}_i^T\mathbf{w}+b=1$ mempunyai bobot $\mathbf{w}$ dan jarak tegak lurus dari titik asal sebesar $\frac{|1-b|}{\|\mathbf{w}\|}$, sedangkan bidang pembatas kedua $\mathbf{x}_i^T\mathbf{w}+b=-1$ mempunyai bobot $\boldsymbol{w}$ dan jarak tegak lurus dari titik asal sebesar $\frac{|-1-b|}{\|\mathbf{w}\|}$. Jarak antara margin dan bidang pemisah (*Separating hyperplane*) adalah $m_+ = m_- = \frac{1}{\|\mathbf{w}\|}$ .Nilai maksmimum margin atau nilai margin (jarak) antara bidang pembatas (berdasarkan rumus jarak garis ketitik pusat) adalah $\frac{1-b-(-1-b)}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}$. (2.7)

Hyperplane yang optimal adalah $\max \frac{2}{\|\mathbf{w}\|}$ atau *equivalent* dengan $\min \frac{1}{2}\|\mathbf{w}\|^2$.

Dengan menggabungkan kedua konstrain pada persamaan (2.6) maka dapat direpresentasikan dalam pertidaksamaan sebagai berikut :

$$y_i(\mathbf{x}_i^T\mathbf{w}+b)-1 \geq 0 \quad , i=1,2,...,n \; . \tag{2.8}$$

Secara matematis, formulasi problem optimasi SVM untuk klasifikasi linier dalam *primal space* adalah

$$\min \frac{1}{2}\|\mathbf{w}\|^2 , \tag{2.9}$$

Dengan fungsi kendala $y_i(\mathbf{x}_i^T\mathbf{w}+b) \geq 1$, $i=1,2,...,$n

Dalam formulasi diatas, ingin meminimalkan fungsi tujuan $\frac{1}{2}\|\mathbf{w}\|^2$ atau sama saja dengan memaksimalkan $\|\mathbf{w}\|^2$ atau $\|\mathbf{w}\|$. Maksimal margin $\frac{2}{\|\mathbf{w}\|}$ dapat diperoleh dari meminimalkan $\|\mathbf{w}\|^2$ atau $\|\mathbf{w}\|$.

Secara umum, persoalan optimasi (2.9) ini akan lebih mudah diselesaikan jika diubah ke dalam formula *langrange.* Dengan demkian permasalahan optimasi dengan konstrain dapat dirumuskan menjadi :

$$L_{\text{pri}}(\mathbf{w},b,\alpha)=\frac{1}{2}\|\mathbf{w}\|^2-\sum_{i=1}^{n}\alpha_i\left\{y_i(\mathbf{x}_i{}^T\mathbf{w}+b)-1\right\}, \tag{2.10}$$

dengan konstrain $\alpha_i \geq 0$ (nilai dari koefisien *lagrange*). Penaksir $\mathbf{w}$ dan b dengan meminimumkan L_{pri} terhadap $\boldsymbol{w}$ dan b dan disamadengankan $\frac{\partial L_{\text{pri}}(\mathbf{w},b,\alpha)}{\partial \mathbf{w}}=\mathbf{0}$ dan $\frac{\partial L_{\text{pri}}(\mathbf{w},b,\alpha)}{\partial b}=0$, sehingga diperoleh Persamaan (2.11)

$$\mathbf{w}=\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \text{ dan } \sum_{i=1}^{n}\alpha_i y_i=0\,. \tag{2.11}$$

Vector $\mathbf{w}$ seringkali bernilai besar (tak terhingga), tetapi nilai α_i terhingga. Untuk itu,formula *langrange* L_{pri} (*primal problem*) diubah ke dalam L_D (*Dual Problem*). Dengan mensubstitusikan persamaan (2.11) ke persamaan (2.10) dieproleh L_D yang ditunjukkan pada persamaan (2.12) :

$$L_D=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i{}^T\mathbf{x}_j\,. \tag{2.12}$$

Jadi persoalan pencarian bidang pemisah terbaik dapat dirumuskan pada persamaan (2.13).

$$\max_{\alpha} L_D=\max\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i{}^T\mathbf{x}_j\,, \tag{2.13}$$

nilai α_i dapat diperoleh, yang nantinya akan digunakan untuk mencari nilai $\mathbf{w}$. Jika nilai $\alpha_i>0$ atau sebuah titik data ke-i untuk setiap $y_i(\mathbf{x}_i{}^T\mathbf{w}+b)=1$. Penyelesaian masalah *primal* dan *dual* pada persamaan (2.10) dan (2.12) memberikan solusi yang sama ketika masalah optimasi adalah *convex*. Setelah menyelesaikan *dual problem*, maka suatu pengamatan baru $\boldsymbol{x}_{(new)}$ dapat diklasifikasikan menggunakan ukuran klasifikasi sebagai berikut:

$$\hat{f}(\mathbf{x}_{new})=sign(\mathbf{x}_{new}{}^T\hat{\mathbf{w}}+\hat{b}), \tag{2.14}$$

dimana $\hat{\mathbf{w}}=\sum_{i=1}^{n}\hat{\alpha}_i y_i \mathbf{x}_i$ dan $\hat{b}=\frac{1}{ns}\left(\sum_{i=1}^{ns}\frac{1}{y_i}-\left(\mathbf{x}_{new}{}^T\hat{\mathbf{w}}\right)\right)$

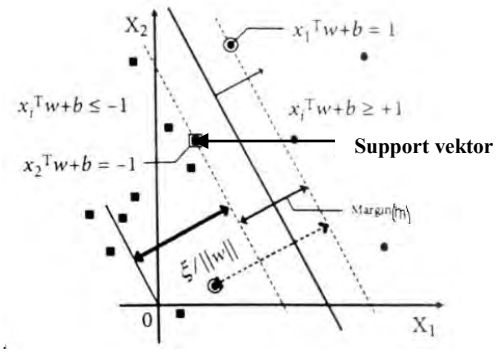
dengan $\mathbf{x}_i$ adalah *support vector,* $\mathbf{x}_{new}$ adalah data yang akan diklasifikasikan, α_i adalah *Langrange Multiplier* dan b adalah bias dan *ns* adalah jumlah *support vector*.

2.3.2 SVM Pada Linier *Nonseparable*

Haerdle, Prastyo dan Hafner (2014) menyatakan pada kasus linier *nonseparable* yaitu mengklasifikasikan data linier yang tidak dapat dipisahkan maka konstrain pada persamaan (2.6) harus diubah secara linier dengan penambahan *variabel slack* ξ_i $(0 \le \xi_i \ge 1, \forall_i)$, sehingga $\mathbf{x}_i$ diklasifikasikan menjadi:

$$\mathbf{x}_i^T \mathbf{w} + b \ge 1 - \xi_i \text{ untuk } y_i = 1 \text{ (untuk kelas +1)}$$

$$\mathbf{x}_i^T \mathbf{w} + b \ge -(1 - \xi_i) \text{ untuk } y_i = -1 \text{ (untuk kelas -1)} \quad (2.15)$$



Gambar 2.6. Bidang pemisah terbaik dengan margin *(m)* terbesar linier non separable (Diperoleh dari Haerdle, 2014)

Bidang pemisah terbaik dengan margin (*m)* terbesar linier *nonseparable*, dapat diilustrasikan pada Gambar 2.6. Pencarian bidang pemisah terbaik dengan penambahan variabel ξ_i sering juga disebut dengan *soft margin hyperplane.* Formula pencarian bidang pemisah terbaik atau fungsi tujuan berubah menjadi :

$$\min_{w,\xi} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\left[\sum_{i=1}^{n} \xi_i\right]. \quad (2.16)$$

Persamaan (2.16) dapat digabungkan ke dalam dua *constraint* dalam bentuk persamaan (2.17) :

$$y_i(\mathbf{x}_i^T \mathbf{w} + b) \ge 1 - \xi_i \quad (2.17)$$

dengan $\xi_i \geq 0,\ C > 0$,

dimana C adalah parameter yang menentukan besar penalti akibat kesalahan dalam klasifikasi (*misclassification*) data dan nilainya ditentukan oleh pengguna. Bentuk persamaan (2.16) memenuhi prinsip *Structural Risk Minimization* (SRM) dimana meminimumkan $\frac{1}{2}\|\mathbf{w}\|^2$ *equivalent* dengan meminimumkan dimensi VC (Vapnik-Chervonenkis). Nilai dari dimensi VC ini akan menentukan besarnya nilai kesalahan hipotesis pada data testing sedangkan meminimumkan $C\sum_{i=1}^{n}\xi_i$ ekuivalen dengan meminimumkan *error* pada data *training*. Fungsi *Langrange* untuk *primal problem* adalah

$$L_{\text{pri}}(\mathbf{w},b,\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left[y_i(\mathbf{x}_i^{T}\mathbf{w}+b)-1+\xi_i\right] - \sum_{i=1}^{n}\mu_i\xi_i \qquad (2.18)$$

dimana $\alpha_i \geq 0$ dan $\mu_i \geq 0$ adalah *Lagrange Multiplier*.

Kondisi KKT (*Karush-Kuhn-Tucker*) untuk *primal problem* adalah :

$$\frac{\partial L_{\text{pri}}(\mathbf{w},b,\alpha)}{\partial \mathbf{w}} = \mathbf{0} \ \rightarrow\ \mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_{\text{i}} = \mathbf{0} \ \Leftrightarrow\ \mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_{\text{i}}$$

$$\frac{\partial L_{\text{pri}}(\mathbf{w},b,\alpha)}{\partial b} = 0 \ \rightarrow\ -\sum_{i=1}^{n}\alpha_i y_i = 0 \ \Leftrightarrow\ \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\frac{\partial L_{\text{pri}}(\mathbf{w},b,\alpha)}{\partial \xi_i} = 0 \ \rightarrow C - \alpha_i - \mu_i = 0 \ \Leftrightarrow C = \alpha_i + \mu_i$$

Mengikuti *constraint* :

$$\xi_i \geq 0,\ \alpha_i \geq 0,\ \mu_i \geq 0,\ \alpha_i\left\{y_i\left(\mathbf{x}_{\mathbf{i}}^{T}\mathbf{w}+b\right)-1+\xi_i\right\} = 0,\ \mu_i\xi_i = 0 \ .$$

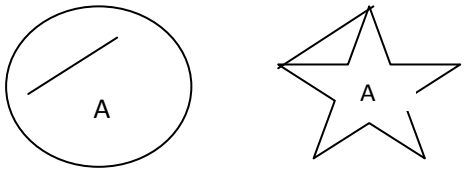
Dengan mensiubstitusikan nilai $\widehat{\boldsymbol{w}} = \sum_{i=1}^{n}\widehat{\alpha}_i y_i \boldsymbol{x}_i$ ke dalam *primal problem* menjadi persamaan *dual problem* sebagai berikut :

$$\max_{\alpha} L_D = \max \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_{\text{i}}\alpha_j \text{y}_{\text{i}}\text{y}_j \mathbf{x}_{\text{i}}^{\text{T}}\mathbf{x}_j \qquad (2.19)$$

dengan $0 \leq \alpha_i \leq C$ dan $\sum_{i=1}^{n}\alpha_i y_i = 0$.

Sampel $\mathbf{x}_i$ untuk $\alpha_i > 0$ (*support vector*) yaitu titik yang berada diatas margin atau dalam margin ketika *Soft margin* digunakan. *Support vector* sering menyebar dan level penyebaran berada pada batas atas (*upper bound*) untuk *misclassification rate* (Secholkopf dan Simola, 2002).

Bentuk persamaan (2.16) merupakan fungsi konveks. Himpunan A dikatakan bersifat konveks jika terdapat dua titik dalam A yang membentuk segmen garis yang terletak dalam A.



Gambar 2.7 Konveks dan Tidak Konveks

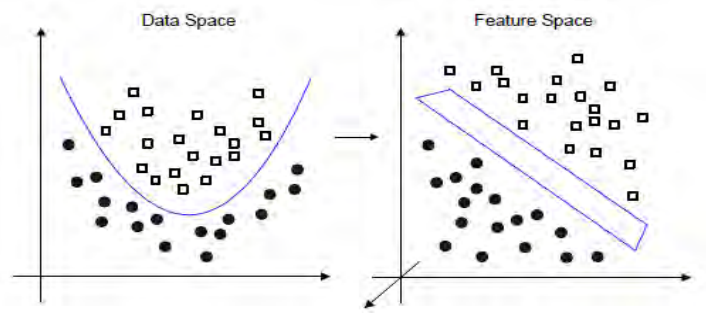
Bentuk kurva yang digambarkan di atas memperlihatkan bentuk konveks dan tidak konveks suatu himpunan sesuai dengan definisi diatas. Untuk masalah optimasi konveks ini menjadi alternatif *dual problem* dari solusi untuk *primal problem* yang diperoleh. Dual problem ini menggantikan sebagai dual variabel langrange *multiplier*. Sebuah himpunan konveks A dalam $\mathbb{R}^p$ didefiisikan sebagai persyaratan untuk dua titik data $\boldsymbol{u}_1 \neq \boldsymbol{u}_2 \in \mathrm{A}$, ini mengikuti $\boldsymbol{u} \in \mathrm{A}$ dengan $\boldsymbol{u} = \lambda \boldsymbol{u}_1 + (1-\lambda)\boldsymbol{u}_2, \forall \lambda \in 0 \leq \lambda \leq 1.$

Sebuah fungsi $f : R^p \rightarrow R$ adalah konveks jika domain f adalah himpunan konveks dan jika untuk setiap $u_1, u_2 \in \text{Domain } f$ dan untuk setiap $0 \leq \lambda \leq 1, f(\lambda u_1 + (1-\lambda)u_2) \leq \lambda f(u_1) + (1-\lambda) f(u_2)$ (Boyd, 2004).

2.3.3 SVM Pada Nonlinier Separable

Menurut Haerdle, Prastyo dan Hafner (2014), pada kenyataan tidak semua data bersifat linier sehingga sulit untuk mencari bidang pemisah secara linier. Diberikan beberapa titik baru $x \in X$ dan ingin memprediksi hubungan $y \in Y = \{-1,1\}$, maksudnya adalah memilih y dimana (x, y) hampir mirip ke *training* sampel. Akhirnya, memerlukan pengukuran kemiripan dalam X dan dalam $\{-1,1\}$ (Chen, Lin dan Scholkopf, 2005). Permasalahn ini dapat diselesaikan

dengan mentransformasikan data ke dalam dimensi ruang yang berdimensi lebih tinggi sehingga dapat dipisahkan secara linier pada *feature space* yang baru. SVM juga bekerja pada data nonlinier.



Gambar 2.8 Mapping dari Dua Dimensi Data *Space* (Kiri) ke Tiga Dimensi *Feature Space* (Kanan) (Diperoleh dari Haerdle, 2014)

Klasifikasi nonlinier yang ditunjukkan pada Gambar 2.7, data dengan sebuah struktur nonlinier fungsi $\varphi : \mathbb{R}^p \to H$ ke dalam *dimensional space* tinggi H dimana pengukuran klasifikasi bersifat linier. Semua *vector training* $\mathbf{x}_i$ dalam persamaan (2.19) berupa *dot product* dengan bentuk $\mathbf{x}_i^T \mathbf{x}_j$. Pada SVM nonlinier, *dot product* ditransformasikan ke $\varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$. Fungsi transformasi pada SVM adalah menggunakan "*Kernel Trick*" (Scholkopf & Simola, 2002). *Kernel Trick* adalah menghitung *scalar product* dalam bentuk sebuah fungsi kernel. Proyeksi $\varphi : \mathbb{R}^p \to H$ memastikan bahwa *inner product* $\varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$ dipresentasikan oleh fungsi kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j) \tag{2.20}$$

Jika sebuah fungsi kernel K pada persamaan (2.20), ini dapat digunakan tanpa perlu mengetahui fungsi transformasi φ secara eksplisit.

Diberikan sebuah kernel K dan data $x_1, x_2, ..., x_n \in X$ maka matrik $K = (K(\mathbf{x}_i, \mathbf{x}_j))_{ij}$ berukuran *nxn* disebut *Gram matrix* untuk data $x_1, x_2, ..., x_n$. Sebuah syarat cukup dan perlu untuk matrik simetri K , dengan $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = K(\mathbf{x}_j, \mathbf{x}_i) = K_{ji}$, untuk K definit postif disebut *"Mercer's Theorem"* (Mercer, 1909).

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\varphi_i\varphi_j K(\mathbf{x}_i,\mathbf{x}_j)\geq 0$$

Contoh sederhana pada sebuah *kernel trick* yang menunjukkan bahwa kernel dapat dihitung tanpa perhitungan fungsi *mapping* φ secara eksplisit adalah fungsi pemetaan :

$$\varphi(\mathrm{x}_1,\mathrm{x}_2)=(\mathrm{x}_1{}^2,\sqrt{2}\mathrm{x}_1\mathrm{x}_2,\mathrm{x}_2{}^2)^T$$

Sehingga menjadi

$$\mathbf{w}^T\ \varphi(\mathrm{x})=\mathrm{w}_1\mathrm{x}_1{}^2+\sqrt{2}\mathrm{w}_2\mathrm{x}_1\mathrm{x}_2+\mathrm{w}_3\mathrm{x}_2{}^2$$

dengan dimensi pada *feature space* adalah kuadratik, padahal dimensi asalnya adalah linier. Metode kernel menghindari pembelajaran secara eksplisit *mapping* data ke dalam *feature space* dimensi tinggi, seperti pada contoh berikut:

$$\begin{aligned}
f(\mathrm{x}) &= \mathbf{w}^T\mathbf{x}+b \\
&= \sum_{i=1}^{n}\alpha_i\mathbf{x}_i{}^T\mathbf{x}+b \\
&= \sum_{i=1}^{n}\alpha_i\varphi(\mathbf{x}_i)^T\varphi(\mathbf{x})+b \text{ dalam feature space } \mathcal{F} \\
&= \sum_{i=1}^{n}\alpha_i\ K(\mathbf{x}_i,\mathbf{x})+b\ .
\end{aligned}$$

Hubungan kernel dengan fungsi *mapping* adalah:

$$\begin{aligned}
\varphi(\mathrm{x}_i)^T\varphi(\mathrm{x}) &= \left(\mathrm{x}_{i1}^2,\sqrt{2}\mathrm{x}_{i1}\mathrm{x}_{i2},\mathrm{x}_{i2}{}^2\right)(\mathrm{x}_1{}^2,\sqrt{2}\mathrm{x}_1\mathrm{x}_2,\mathrm{x}_2{}^2)^T \\
&= \mathrm{x}_{i1}^2\mathrm{x}_1^2+2\mathrm{x}_{i1}\mathrm{x}_{i2}\mathrm{x}_1\mathrm{x}_2+\mathrm{x}_{i2}^2\mathrm{x}_2^2 \\
&= (\mathbf{x}_i^T\mathbf{x})^2 \\
&= K(\mathbf{x}_i,\mathbf{x})
\end{aligned}$$

Sedangkan, untuk memperoleh fungsi klasifikasi nonlinier dalam data *space,* bentuk secara umumnya dipeoleh dari penerapan *kernel trick* ke persamaan (2.21):

$$L_D=\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)\ , \tag{2.21}$$

yaitu memaksimumkan $L_D:\max_{\alpha}L_D=\max\sum_{i=1}^{n}\alpha_i-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)$

dengan $\sum_{i=1}^{n}\alpha_i y_i=0$, $0\leq\alpha_i\leq C$; i=1,2,...,n

Fungsi kernel yang umum digunakan pada metode SVM adalah :

1. Kernel Linier

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\mathbf{x}_{\mathrm{i}}^{\mathrm{T}}\mathbf{x}_j$$

2. Kernel Polynomial

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\left(\delta\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j+r\right)^p,\delta>0$$

3. Kernel Radial basis function (RBF)

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\exp\left(-\frac{\left\|\mathbf{x}_i-\mathbf{x}_j\right\|^2}{2\sigma^2}\right)$$

4. Kernel sigmoid

$$K\left(\mathbf{x}_i,\mathbf{x}_j\right)=\tanh\left(\delta\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j+r\right)$$

Pemilihan fungsi kernel yang tepat merupakan hal yang sangat penting karena akan menentukan *feature space* dimana fungsi *classifier* akan dicari. Sepanjang fungsi kernelnya sesuai (cocok), SVM akan beroperasi secara benar meskipun tidak tahu pemetaan yang digunakan (Santosa, 2007; Robandi, 2008). Menurut Scholkopf dan Simola (1997), fungsi kernel gaussian RBF memilki kelebihan yaitu secara otomatis menentukan nilai, lokasi dari *center* dan nilai pembobot dan bisa mencakup nilai rentang tak terhingga. Gaussian RBF juga efektif menghindari *overfitting* dengan memilih nilai yang tepat untuk parameter C dan σ dan RBF baik digunakan ketika tidak ada pengetahuan terdahulu. Menurut Hsu, Chang dan Lin (2004), fungsi kernel yang direkomendasikan untuk diuji pertama kali adalah fungsi kernel RBF karena dapat memetakan hubungan tidak linier RBF lebih robust terhadap outlier karena fungsi kernel RBF berada antara selang $(-\infty,\infty)$ sedangkan fungsi kernel yang lain memiliki rentang antara (-1 sampai dengan 1).

2.4 Least Square Support Vector Machine (LS-SVM)

Suyken dan Vandewalle (1999) mengusulkan sebuah versi *least squares* untuk algoritma pembelajaran *Support Vector Machine* (SVM) yang disebut *Least Squares Support Vector Machine* (LS-SVM). LS-SVM adalah modifikasi metode SVM standar yang mengarah pada pemecahan linier sistem *Karush-Kuhn-Tucker* (KKT). Dalam formulasi LS-SVM, perhitungan komputasi dari SVM yang disederhanakan dengan pelaksanaan versi *Least Squares* (LS) daripada *inequality constraints* dan fungsi penalti penjumlahan kesalahan kuadrat (*squared error*) sebagaimana digunakan dalam pelatihan jaringan saraf tiruan. Reformulasi ini sangat menyederhanakan masalah dalam memecahkan satu set persamaan linier daripada pemrograman kuadratik (*quadratic programming*) yang digunakan dalam SVM standar. *Primal problem* pada LS-SVM atau fungsi tujuan dirumuskan dengan memodifikasi persamaan (2.16) sebagai berikut :

$$\min_{w,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{2}C\sum_{i=1}^{n}\xi_i^2 \tag{2.22}$$

dengan $y_i[\varphi(\mathbf{x}_i)^T\mathbf{w}+b] = 1-\xi_i$; $i = 1,....,n$ dan

fungsi kernel $\Omega_{ij} = K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \varphi\left(\mathbf{x}_i\right)^{\mathrm{T}}\varphi\left(\mathbf{x}_j\right)$

Fungsi Lagrange dari persamaan (2.22) :

$$L_{\mathrm{pri}}(\mathbf{w},b,\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 + C\frac{1}{2}\sum_{i=1}^{n}\xi_i^2 - \sum_{i=1}^{n}\alpha_i\left(y_i[\varphi(\mathbf{x}_i)^T\mathbf{w}+b]-1+\xi_i\right) \tag{2.23}$$

dengan α_i adalah pengali *lagrange* (dapat bernilai positif atau negative). Persamaan (2.23) dengan $\mathbf{w}$,b dan α_i untuk kondisi optimal dapat digambarkan sebagai berikut :

$$\begin{cases} \dfrac{\partial L_{pri}(\mathbf{w},b,\alpha)}{\partial \mathbf{w}} = \mathbf{0} \rightarrow \mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \varphi(\mathbf{x}_i) \\ \dfrac{\partial L_{pri}(\mathbf{w},b,\alpha)}{\partial b} = 0 \rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0 \\ \dfrac{\partial L_{pri}(\mathbf{w},b,\alpha)}{\partial \xi_i} = 0 \rightarrow \alpha_i = C\xi_i, \quad i = 1,....,n \\ \dfrac{\partial L_{pri}(\mathbf{w},b,\alpha)}{\partial \alpha_i} = 0 \rightarrow y_i[\varphi(\mathbf{x}_i)^T\mathbf{w}+b] = 1-\xi_i, \quad i = 1,...,n \end{cases} \tag{2.24}$$

dapat ditulis sebagai sistem linier sebagai ganti dari *Quadratic Programming* sebagai berikut :

$$\left[\begin{array}{ccc|c} \mathbf{I} & 0 & 0 & -\mathbf{Z}^{\mathrm{T}} \\ 0 & 0 & 0 & -\mathbf{y}^{\mathrm{T}} \\ 0 & 0 & \mathrm{C}\mathbf{I} & -\mathbf{I} \\ \hline \mathbf{Z} & \mathbf{y} & \mathbf{I} & 0 \end{array}\right]\left[\begin{array}{c} \mathbf{w} \\ b \\ \xi \\ \hline \alpha \end{array}\right]=\left[\begin{array}{c} 0 \\ 0 \\ 0 \\ \hline \mathbf{1} \end{array}\right] \tag{2.25}$$

Dengan

$\mathbf{Z}=[\varphi(\mathbf{x}_i)^T y_1 ,\ldots, \varphi(\mathbf{x}_n)^T y_n]^T$, $\mathbf{y}=\left[y_1, \mathrm{y}_2\ \ldots, y_n\right]^T$, $\mathbf{1}=[1,1,...,1]^T$, $\xi=[\xi_1,...,\xi_n]^T$,

$\alpha=[\alpha_i,\ldots,\alpha_n]^T$, $e=\left[e_1,\ \ldots,e_n\right]^T$, C adalah *regularization* parameter atau ongkos penalti akibat *misclassification* dan **I** adalah matrik identitas. Setelah eliminasi **w** dan ξ mengikuti sistem linier Karush-Kuhn Tucker (KKT) sehingga menghasilkan Persamaan (2.26)

$$\left[\begin{array}{c|c} 0 & \mathbf{y}^{\mathrm{T}} \\ \hline \mathbf{y} & \mathbf{Z}\mathbf{Z}^{\mathrm{T}}+\mathrm{C}^{-1}\mathbf{I} \end{array}\right]\left[\begin{array}{c} \mathrm{b} \\ \hline \alpha \end{array}\right]=\left[\begin{array}{c} 0 \\ \hline \mathbf{1} \end{array}\right]$$

$$\left[\begin{array}{c|c} 0 & \mathbf{y}^{\mathrm{T}} \\ \hline \mathbf{y} & \Omega+\mathrm{C}^{-1}\mathbf{I} \end{array}\right]\left[\begin{array}{c} \mathrm{b} \\ \hline \alpha \end{array}\right]=\left[\begin{array}{c} 0 \\ \hline \mathbf{1} \end{array}\right] \tag{2.26}$$

Kondisi Mercer diaplikasikan ke matriks $\Omega=\mathbf{Z}\mathbf{Z}^{\mathrm{T}}$ dan kernel *trick* diaplikasikan dalam matrik Ω

$$\begin{aligned} \Omega_{ij} &= y_i y_j \varphi(\mathbf{x}_i)^{\mathrm{T}} \varphi(\mathbf{x}_j) \\ &= y_i y_j\, K(\mathbf{x}_i, \mathbf{x}_j)\ ; \quad i\,, j=1,2,...,\mathrm{n} \end{aligned}$$

2.5 Klasifikasi LS-SVM Multiclass One Against One (OAO)

Menurut Suyken dan Vandewalle (1999c), pendekatan metode OAO diperlukan untuk menemukan fungsi pemisah sebanyak *v(v − 1) /2*, dimana masing-masing fungsi pemisah di*training* dengan sampel dari dua kelas. Misalkan, terdapat persoalan klasifikasi dengan 3 kelas berarti dapat ditentukan 3 fungsi pemisah *v* yaitu v^{12}, v^{13}, dan v^{23}. Ketika v^{12} di*training*, semua sampel pada kelas 1 diberi label positif (+1) dan semua sampel pada kelas 2 diberi label *negative* (-1). Hal ini juga dilakukan pada v^{13} dan v^{23}. Sebagai gambaran, misalkan

terdapat data training sebanyak n yaitu $(\mathrm{x}_1, y_1),\ldots,(\mathrm{x}_n, y_n)$ dimana $\mathbf{x}_i \epsilon\, R,\ i = 1,\ 2,...,n$ adalah data input dan $\mathrm{y_i} \epsilon$ R, k,l = 1, …, v kelas dari $\mathbf{x}_i$ yang bersangkutan maka optimasi penyelesaiannya adalah.

$$\min_{\mathbf{w}^{kl},b^{kl},\xi^{kl}} \frac{1}{2}\left(\mathbf{w}^{kl}\right)^T \mathbf{w}^{kl} + \frac{1}{2} C \sum_{i=1}^{n} \xi_i^{2^{kl}} \tag{2.27}$$

dengan,

$\varphi(\mathbf{x}_i)^{\mathrm{T}} \mathbf{w}^{kl} + b^{kl} = 1 - \xi_i^{kl}$, jika $y_i = k$

$\varphi(\mathbf{x}_i)^T \mathbf{w}^{kl} + b^{kl} = -1 + \xi_i^{kl}$, jika $y_i = l$

$\xi_i^{kl} \geq 0$

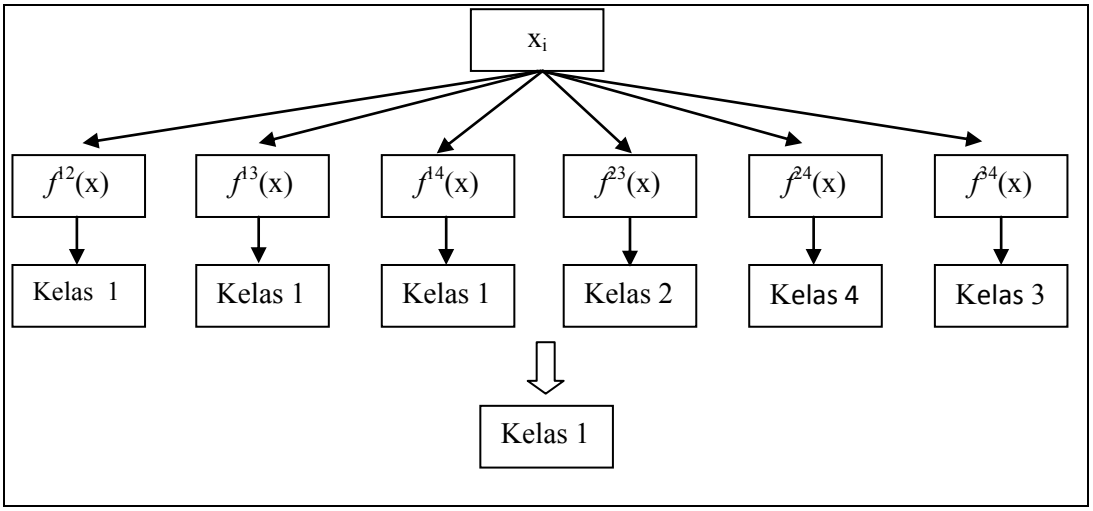
Setelah semua fungsi pemisah *v(v − 1) /2* ditemukan, ada beberapa metode untuk melakukan *testing* pada data baru. Salah satu strategi adalah *max voting* (Santosa, 2013). Berdasarkan pada strategi *max voting*, jika data **x** dimasukkan ke dalam fungsi hasil *training* dan hasilnya menyatakan **x** adalah kelas *k*, maka suara untuk kelas *k* ditambah satu. Kelas dari data **x** akan ditentukan dari jumlah *voting* terbanyak. Jika terdapat dua buah kelas yang jumlah *voting* sama, maka kelas yang indeksnya lebih kecil dinyatakan sebagai kelas dari data. Persamaan *construct* mengikuti *hyperplane* sebagai berikut.

$$\hat{f}^{kl}\left(\mathbf{x}\right) = sign(\hat{\mathbf{w}}^{kl^T}\mathbf{x}) + b^{kl} = 0 \tag{2.28}$$

Contohnya, terdapat permasalahan klasifikasi dengan 4 buah kelas. Oleh karena itu, digunakan 6 buah LS-SVM biner seperti pada Tabel 2.4 dan contoh penggunaanya dalam memprediksi kelas data baru dapat dilihat pada Gambar 2.9.

Tabel 2.4 Ilustrasi *One Against One* (OAO)

$y_{i=1}$	$y_{i=-1}$	Model
Kelas 1	Kelas 2	$\hat{f}^{12}(\mathrm{x}) = \left(\hat{\mathbf{w}}^{12}\right)^T \mathbf{x} + b^{12}$
Kelas 1	Kelas 3	$\hat{f}^{13}(\mathrm{x}) = \left(\hat{\mathbf{w}}^{13}\right)^T \mathbf{x} + b^{13}$
Kelas 1	Kelas 4	$\hat{f}^{14}(\mathrm{x}) = \left(\hat{\mathbf{w}}^{14}\right)^T \mathbf{x} + b^{14}$
Kelas 2	Kelas 3	$\hat{f}^{23}(\mathrm{x}) = \left(\hat{\mathbf{w}}^{23}\right)^T \mathbf{x} + b^{23}$
Kelas 2	Kelas 4	$\hat{f}^{24}(\mathrm{x}) = \left(\hat{\mathbf{w}}^{24}\right)^T \mathbf{x} + b^{24}$
Kelas 3	Kelas 4	$\hat{f}^{34}(\mathrm{x}) = \left(\hat{\mathbf{w}}^{34}\right)^T \mathbf{x} + b^{34}$



Gambar 2.9 Ilustrasi *One Against One* (OAO) (Diperoleh dari Trapsilasiwi, 2013)

2.6 Optimasi Parameter Particle Swarm Optimization (PSO)-Gravitational Search Algorithm (PSO-GSA)

Terdapat dua parameter yang digunakan dalam seleksi parameter C pada SVM dan parameter kernel σ yang menunjukkan *non linier mapping* dari *input space* menjadi *feature space* berdimensi tinggi atau parameter *bandwidth*. Studi empiris sebelumnya (Huang, dkk.,2007) menunjukkan bahwa C optimal terletak

pada *range* antara 10^0 dan 10^2. Parameter kernel σ bergantung pada jarak antara dua titik data.

Teknik *Particle Swarm Optimization* (PSO) dikemukakan oleh Eberhart dan Kennedy (1995). Pada PSO terdapat nilai batasan yang berisi nilai C dan σ kemudian tiap partikel memiliki posisi $\theta(C,\sigma)_i^p=(\theta_i^1,\theta_i^2)$, i=1,2,...,n , $\theta(C,\sigma)_i^p$ merepresentasikan nilai parameter C dan σ. Kecepatan $\boldsymbol{V}_i=(V_i^1,V_i^2)$ pada ruang pencarian berdimensi dua, dimana *i* menyatakan partikel ke-*i* partikel menyatakan penaksir $\theta(C,\sigma)$.

Model dari PSO terdiri dari sekumpulan partikel yang diinisialisasi dengan populasi dari kandidat solusi secara acak. Partikel bergerak melalui ruang masalah *p*-dimensi untuk mencari solusi baru, dengan *fitness* (kelayakan), *fitness* dapat dihitung sebagai ukuran kebaikan solusi yang pasti atau dapat didefinisikan sebagai rata-rata akurasi klasifikasi selama q-*fold cross validation* (Kennedy, Eberhart dan Shi, 2001). Setiap iterasi masing-masing partikel memperbarui posisinya mengikuti dua nilai terbaik, yaitu solusi terbaik yang telah didapat oleh masing-masing partikel atau *local best* disebut "*Pbest*", dinyatakan dalam vektor $\boldsymbol{Pbest}_i=\left(Pb_i^1,Pb_i^2,...,Pb_i^p\right)$ dan solusi terbaik pada populasi atau *global best* diantara kumpulannya (*swarm*) disebut "*Gbest*", dinyatakan dalam vektor $\boldsymbol{Gbest}=\left(Gb^1,Gb^2...,Gb^p\right)$. Kecepatan V_i pada iterasi ke-t diperbaruhi pada iterasi selanjutnya menggunakan persamaan (2.29). Posisi yang baru ditentukan oleh penjumlahan dari posisi sebelumnya dan kecepatan baru yang ditunjukkan pada Persamaan (2.30).

$$\boldsymbol{V}_i^p(t+1)=\omega\boldsymbol{V}_i^p(t)+c_1r\times\left(\boldsymbol{Pbest}_i^p(t)-\theta_i^p(t)\right)+c_2r\times\left(\boldsymbol{Gbest}^p(t)-\theta_i^p(t)\right) \qquad (2.29)$$

$$\theta_i^p(t+1)=\theta_i^p(t)+\boldsymbol{V}_i^p(t+1) \qquad (2.30)$$

$$\omega=w_{\min}+\left(w_{\max}-w_{\min}\right)\frac{(T*-t*)}{T*}$$

dengan

$\boldsymbol{V}_i^p(t)$	=	Kecepatan individu *i* pada iterasi *ke-t* dalam dimensi *p*
ω	=	Bobot inersia
c_1, c_2	=	konstanta positif yang diboboti.
r	=	bilangan random antara 0 dan 1 dari distribusi uniform untuk keragaman pergerakan partikel $r \sim U(0,1)$
$\theta_i^p(t)$	=	Solusi (posisi) individu *i* pada iterasi *ke-t* dalam dimensi *p*
$\boldsymbol{Pbest}_i^p(t)$	=	P*best* individu *i* sampai iterasi *ke-t* dalam dimensi *p*
$\boldsymbol{Gbest}^p(t)$	=	G*best* kelompok sampai iterasi *ke-t* dalam dimensi *p*
w_{min}, w_{max}	=	bobot awal dan akhir
T^*	=	jumlah iterasi maksimum
t^*	=	jumlah iterasi sekarang
t	=	Iterasi

Gravitational Search Algorithm (GSA) merupakan metode optimasi heuristik baru yang diusulkan oleh Rashedi pada tahun 2009. Teori GSA terinspirasi dari teori Newton. Teori yang menyatakan bahwa setiap partikel di alam semesta menarik setiap partikel lain dengan kekuatan yang berbanding lurus dengan perkalian massa partikel dan berbanding terbalik dengan kuadrat dari jarak antar partikel (Newton, 1729).

GSA secara matematis dimodelkan sebagai berikut. Misalkan suatu sistem dengan jumlah n agen. Algoritma ini dimulai dengan menempatkan semua agen secara acak di ruang pencarian. Selama iterasi, gaya gravitasi dari agen *j* terhadap agen *i* pada waktu tertentu didefinisikan sebagai berikut (Rashedi, 2009).

$$\boldsymbol{F}_{ij}^p(t) = G(t)\frac{MGP_i(t) \times MGA_j(t)}{d_{ij}(t) + \varepsilon}\left(\theta_j^p(t) - \theta_i^p(t)\right) \tag{2.31}$$

dengan, MGA_j adalah massa gravitasi aktif yang berhubungan dengan agen *j*, MGP_i adalah massa gravitasi pasif yang berhubungan dengan agen *i,* G(t) adalah konstanta gravitasi pada iterasi ke-t, ε merupakan konstanta yang bernilai sangat kecil, dan d_{ij}(t) merupakan jarak Euclidean antara dua agen yaitu agen *i* dan *j*.

Nilai G(t) diperoleh dari,

$$G(t) = G_0 \times \exp(-\beta \times \frac{t^*}{T^*}) \tag{2.32}$$

dengan β dan G_0 adalah koefisien penurunan dan nilai awal, t^* merupakan jumlah iterasi sekarang serta T^* adalah jumlah maksimum dari iterasi.

Pada kondisi permasalahan dalam dimensi *p*, gaya total yang bekerja pada agen *i* pada iterasi ke-t dihitung dengan persamaan sebagai berikut.

$$F_i^p(t) = \sum_{j=1, j\neq i}^{n} r(F_{ij}^p(t)) \tag{2.33}$$

dengan $r \sim U(0,1)$

Berdasarkan hukum gerak, percepatan agen sebanding dengan kekuatan hasil dan *invers massanya*, sehingga percepatan semua agen harus dihitung sebagai berikut.

$$ac_i^p(t) = \frac{F_i^p(t)}{M_i(t)} \tag{2.34}$$

dengan *t* adalah waktu tertentu atau iterasi dan M_i adalah massa dari objek *i*.

Algoritma PSO-GSA dikembangkan oleh Mirjalili (2012), untuk optimasi parameter sehingga menghasilkan solusi terbaik. Ide dasar dari PSO-GSA adalah untuk menggabungkan kemampuan global terbaik (*Gbest*) algoritma PSO dengan kemampuan pencarian lokal (*Pbest*)pada algoritma GSA.

Dalam rangka untuk menggabungkan algoritma ini, maka algoritma tersebut dapat diformulasikan menggunakan Persamaan (2.35).

$$\boldsymbol{V}_i^p(t+1) = \omega \times \boldsymbol{V}_i^p(t) + c_1 r \times ac_i^p(t) + c_2 r \times \left(\boldsymbol{Gbest}^p(t) - \theta_i^p(t)\right) \tag{2.35}$$

dengan $\boldsymbol{V}_i^p(t)$ adalah kecepatan dari agen *i* pada iterasi *t*, c_1, c_2 adalah konstanta positif yang diboboti, ω adalah bobot inersia, *r* adalah bilangan random diantara 0 dan 1, $ac_i^p(t)$ adalah percepatan agen *i* pada iterasi *t*, dan *Gbest* adalah solusi global terbaik. Pada masing-masing iterasi, posisi partikel di perbarui (*update*) sebagai berikut.

$$\theta_i^p(t+1) = \theta_i^p(t) + \boldsymbol{V}_i^p(t+1) \tag{2.36}$$

Dalam implemmentasi optimasi parameter PSO-GSA, setiap posisi $\theta_i^p(t)$ merepresentasikan nilai parameter LS-SVM σ dan C. Dalam PSO-GSA, pada proses awal/inisialisasi, semua agen diinisialisasi secara acak. Setiap agen dianggap sebagai kandidat solusi permasalahan. Setelah proses inisialisasi maka gaya gravitasi, konstanta gravitasi, dan resultan gaya antara agen dihitung dengan menggunakan persamaan (2.31), (2.32), dan (2.33). Setelah itu, percepatan partikel dihitung dengan persamaan (2.34). Dalam setiap iterasi, solusi yang terbaik selalu diperbarui. Setelah menghitung percepatan dan memperbarui solusi yang terbaik, kecepatan dari semua agen dapat dihitung menggunakan persamaan (2.35). Akhirnya, posisi agen dihitung menggunakan persamaan (2.36). Proses memperbarui kecepatan dan posisi akan dihentikan dengan memenuhi kriteria akhir.

2.7 Evaluasi Performansi Metode Klasifikasi

Data actual dan data hasil prediksi dari model klasifikasi disajikan dengan menggunakan Tabulasi silang (*Confusion matrix*), yang mengandung informasi tentang kelas data yang actual direpresentasikan pada baris matriks dan kelas data hasil prediksi pada kolom (Han. Jiawei,dkk, 2006).

Tabel 2.5 Confusion Matrix

Pengelompokan Aktual	Kelompok Prediksi					Total
	1	2	3		k	
1	X_{11}	X_{12}	X_{13}	...	X_{1k}	n_1
2	X_{21}	X_{22}	X_{23}	...	X_{2k}	n_2
3	X_{31}	X_{32}	X_{33}	...	X_{3k}	
⋮				⋮		⋮
k	X_{k1}	X_{k2}	X_{k3}	...	X_{kk}	n_k
Total	n_1	n_2	n_3	...	n_k	N_{total}

(Akbar, Yudhistira dan Cholissodin, 2014).

Keterangan :

$$TP = X_{11} + X_{22} + \cdots + X_{kk}$$

$$TN = (X_{11} + X_{22}) + (X_{11} + X_{33}) + (X_{22} + X_{33}) \cdots + (X_{11} + X_{kk}) + (X_{22} + X_{kk}) + (X_{33} + X_{kk})$$

$$FP = (X_{21} + X_{31} + \ldots + X_{k1}) + (X_{12} + X_{32} + \ldots + X_{k2}) + (X_{13} + X_{23} + \ldots + X_{k3}) + \ldots + (X_{k1} + X_{k2} + \ldots + X_{1k})$$

$$FN = (X_{12} + X_{13} + \ldots + X_{1k}) + (X_{21} + X_{23} + \ldots + X_{2k}) + (X_{31} + X_{32} + \ldots + X_{3k}) + \ldots + (X_{1k} + X_{2k} + \ldots + X_{k1})$$

1. *True Postive* (TP) menunjukan bahwa kelas yang dihasilkan prediksi klasifikasi adalah positif dan kelas sebenarnya adalah positif
2. *True Negatif* (TN) menunjukan bahwa kelas yang dihasilkan dari prediksi klasifikasi adalah negatif dan kelas sebenarnya adalah negatif.
3. *False Positif* (FP) menunjukan bahwa kelas yang dihasilkan dari prediksi klasifikasi adalah negatif dan kelas sebenarnya adalah positif
4. *False Negatif* (FN) menunjukan bahwa kelas yang dihasilkan dari prediksi klasifikasi adalah positif dan kelas sebenarnya adalah negatif.

Ketepatan klasifikasi dapat dilihat dari akurasi klasifikasi. Akurasi klasifikasi menunjukkan performansi model klasifikasi secara keseluruhan, dimana semakin tinggi akurasi klasifikasi hal ini berarti semakin baik performansi model klasifikasi.

$$\text{Akurasi Total} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah total prediksi}} \times 100\%$$

$$\text{Akurasi Total} = \frac{X_{11} + X_{22} + \cdots + X_{kk}}{N_{total}} \times 100\% \quad (2.37)$$

Untuk mendapatkan klasifikasi yang optimal dan lebih spesifik maka dapat diuji *Sensitivity* dan *Specificity*. *Sensitivity* adalah tingkat positif benar atau ukuran performansi untuk mengukur kelas yang positif (minor) sedangkan *Specificity* adalah tingkat negative benar atau ukuran performansi untuk mengukur kelas yang negatif (mayor). Rumus *Sensitivity* dan *Specificity* adalah sebagai beirkut.

$$Sensitivity = \frac{TP}{(TP + FN)} \times 100\% \quad (2.38)$$

$$Specificity = \frac{TN}{(TN + FP)} \times 100\% \quad (2.39)$$

Selain itu, evaluasi performansi model klasifikasi dapat dilakukan dengan menggunakan G-mean dan F-measure. Berikut ini merupakan penjelasan tentang G-Mean dan F-Measure,

G-Mean merupakan rata-rata *geometrik Sensitivity* dan *Specificity*. Apabila semua kelas psoitif tidak dapat diprediksi maka G-Mean akan bernilai nol sehingga diharapkan suatu algoritma klasifikasi mencapai nilai G-Mean yang tinggi (Kubat dan Matwin dalam Sain, 2013). Dengan rumus berikut ini.

$$G-Mean=\sqrt{Sensitivity\times Specificity} \tag{2.40}$$

Pengukuran akurasi dari *imbalanced* class dapat dilakukan dengan menggunakan perhitungan nilai *recall, precision* dan *f-measure*. Recall dihitung untuk mengevaluasi seberapa *coverage* suatu model dalam memprediksi suatu kelas tertentu yaitu kelas positif (minor). Nilai *recall* sama dengan nilai *Sensitivity*. *Precision* dihitung untuk mengevaluasi seberapa baik ketepatan model dalam memprediksi suatu kelas positif. Nilai *F-measure* dihitung untuk menentukan hasil prediksi yang paling baik, yang merupakan kombinasi dari nilai *recall* dan *precision*. Dengan rumus berikut (Cao dkk dalam Sain, 2013).

$$Recall\ /\ Sensitivity=\frac{\text{kategori ditemukan benar}}{\text{Total kategori ditemukan}}=\frac{\text{TP}}{(\text{TP}+\text{FN})}\times 100\% \tag{2.41}$$

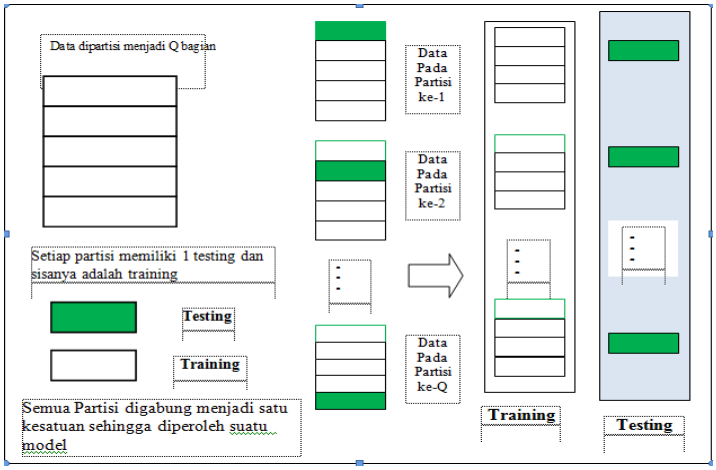
$$Precision=\frac{\text{kategori ditemukan benar}}{\text{Total kategori benar}}=\frac{\text{TP}}{(\text{TP}+\text{FP})}\times 100\% \tag{2.42}$$

$$F\text{ - }Measure=\frac{2\times\left(Sensitivity\times Precision\right)}{\left(Sensitivity+Precision\right)}\times 100\% \tag{2.43}$$

2.8 Q - Fold Cross Validation

Cross-Validation adalah pembagian data *training* dan data *testing* secara random yang dilakukan dengan menggunakan q-*fold cross validation*. Q-fold *cross validation* akan membagi data ke dalam (*q) subset* yang saling bebas yaitu $S_1, S_2, ..., S_q$ dengan jumlah data setiap *subset* yang hampir sama, selanjutnya jika satu *subset* menjadi data *testing* maka (*q*-1) subset yang akan menjadi data *training* (Han. Jiawei,dkk, 2006). Nilai *cross validation error estimate* pada fold atau partisi (1,2,…,Q) adalah $CV_{(Q)}=\sum_{q=1}^{Q}\frac{n_q}{n}(1-Akurasi\ Total)_q$

dengan n_q adalah banyaknya data pada partisi ke-q dan n adalah banyaknya data keseluruhan.



Gambar 2.11 Ilustrasi Pembagian Data Training dan Testing dengan Q Fold

2.9 Uji Friedman

Menurut Daniel (1989), Uji Friedman analog dengan analisis *two way anova* pada parametrik. Pengujian dilakukan terhadap tiga atau lebih kelompok. Asumsi uji tersebut sama seperti uji nonparametrik yang lain yaitu

1. Data terdiri atas *b* buah sampel (blok) berukuran *k* yang saling bebas. Nilai perlakuan ke-*j* dalam sampel atau blok ke-*i* disebut $X_{ij.}$ Struktur data diilustrasikan seperti Tabel 2.6, dengan baris-baris untuk blok-blok (*dataset*) dan kolom-kolom untuk perlakuan-perlakuan (treatment atau metode).
2. Tidak ada interaksi antara blok-blok dan perlakuan-perlakuan
3. Nilai- nilai pengamatan dalam masing-masing blok boleh diperingkat menurut besarnya.

Hipotesis :

H_0 : $R_1 = R_2 = ... = R_k$

H_1 : minimal ada satu dari R_j berbeda atau tidak sama ; (j=1,2,...,k)

Struktur data yang digunakan dalam uji Friedman , seperti pada Tabel 2.6.

Tabel 2.6 Struktur Data Uji Friedman

Blok (*dataset*)	Perlakuan (metode) (*j*)						Perlakuan (metode) (*j*)					
	1	2	...	*j*	...	*k*	1	2	...	*j*	...	*k*
1	$X_{1,1}$	$X_{i,1}$	...	$X_{i,j}$	...	$X_{i,k}$	$R_{1,1}$	$R_{1,2}$	...	$R_{1,j}$	...	$R_{1,k}$
2	$X_{2,1}$	$X_{2,2}$	...	$X_{2,j}$	...	$X_{2,k}$	$R_{2,1}$	$R_{2,2}$	...	$R_{2,j}$	...	$R_{2,k}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
i	$X_{i,1}$	$X_{i,2}$	...	$X_{i,j}$	...	$X_{i,k}$	$R_{i,1}$	$R_{i,2}$	...	$R_{i,j}$	...	$R_{i,k}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
b	$X_{b,1}$	$X_{b,2}$	...	$X_{b,j}$	...	$X_{b,k}$	$R_{b,1}$	$R_{b,2}$	...	$R_{b,j}$	...	$R_{b,k}$
Jumlah peringkat (R)							R_1	R_2	...	R_j	...	R_k

Untuk perhitungan, nilai-nilai pengamatan dalam masing-masing blok diperingkat dari yang terkecil hingga terbesar secara terpisah, sehingga masing-masing blok memiliki *k* buah peringkatnya sendiri-sendiri.

Statistik Uji :

$$\chi^2 = \frac{12}{bk(k+1)} \sum_{J=1}^{k} R_j^2 - 3b(k+1) \tag{2.46}$$

Jika ada cukup banyak angka yang sama (*ties*) maka statistik uji yang digunakan adalah :

$$\chi_c^2 = \frac{\chi^2}{1 - \frac{\sum_{i=1}^{b} t_i^3 - \sum_{i=1}^{b} t_i}{bk(k^2-1)}} \tag{2.47}$$

dengan $T_i = \sum_{i=1}^{b} t_i^3 - \sum_{i=1}^{b} t_i$ dimana T_i adalah banyaknya nilai pengamatan yang sama untuk suatu peringkat dalam blok ke-*i*.

R_j adalah jumlah peringkat dari perlakuan ke-*j* (*j*=1,2,...,*k*), *b* adalah banyaknya blok (*dataset*) dan *k* adalah banyaknya perlakuan (metode).

Daerah Kritis :Tolak H_0 jika $\chi^2_{hitung} > \chi^2_{\alpha,k-1}$ atau $\chi^2_{c(hitung)} > \chi^2_{\alpha,k-1}$ atau p-Value<α

2.10 Uji Perbandingan Berganda

Apabila terjadi keadaan tolak H_0 pada uji Friedman, maka untuk mengetahui perlakuan (metode) mana yang berbeda maka dilanjutkan dengan uji perbandingan berganda (Daniel, 1989).

Hipotesis :

H_0 : $R_j = R_{j^*}$ (tidak terdapat perbedaan efek perlakuan j dengan j*)

H_1 : $R_j \neq R_{j^*}$ (terdapat perbedaan efek perlakuan j dengan j*)

Statistik Uji: $\left|R_j - R_{j^*}\right|$ (2.48)

Daerah Kritis : Tolak H_0 apabila $\left|R_j - R_{j^*}\right| > Z_{\{1-(\alpha/k(k-1))\}}\sqrt{\frac{bk(k+1)}{6}}$

dimana R_j adalah jumlah peringkat dari perlakuan ke-j (j=1,2,...,k), k adalah banyaknya perlakuan (metode) dan b adalah banyaknya blok (*dataset*).

2.11 Uji Dua Sampel Independen Mann Whitney

Pengujian dua sampel independen Mann Whitney adalah sebagai berikut.

Hipotesis

H_0 : $R_1 = R_2$

H_1 : $R_1 \neq R_2$

Statistik Uji : $Z = \dfrac{W - n_1 n_2 / 2}{\sqrt{n_1 n_2 (n_1 + n_2)/12}}$ (2.49)

dimana $W = \widehat{R}_1 - \dfrac{n_1(n_1+1)}{2}$

R_1 adalah jumlah rangking pada kelompok pertama, n_1, n_2 adalah jumlah sampel pada kelompok pertama dan kedua.

Bila ada angka-angka sama cukup banyak, dilakukan koreksi dengan :

$$Z = \frac{W - n_1 n_2 / 2}{\sqrt{n_1 n_2 (n_1 + n_2)/12 - \dfrac{n_1 n_2 \left(\sum_{i=1}^{b} t_i^3 - \sum_{i=1}^{b} t_i\right)}{12(n_1 + n_2)(n_1 + n_2 - 1)}}} \quad (2.50)$$

Daerah Kritis : Tolak H_0 jika $Z_{hitung} > Z_{\alpha/2}$ atau $Z_{hitung} < -Z_{\alpha/2}$ atau p-Value$< \alpha$

2.12 Penelitian Sebelumnya

Penelitian sebelumnya tentang *imbalanced* data adalah sebagai berikut

Tabel 2.7 Daftar Penelitian Sebelumnya

Peneliti, Tahun	**Ringkasan**
Kubat dan Matwin, 1997	Menghapus kasus borderline pada data kelas negatif dengan metode *Tomek Links*
Ling dan Li, 1998	Menduplikasi data kelas positif dengan metode *oversampling*
Chawla, 2002	Mereplikasi data kelas positif dengan metode *Synthetic Minority Oversampling Technique* (SMOTE)
Batista dkk,2003 dan 2004	Menggunakan metode SMOTE+Tomek Links dengan klasifikasi decision Tree
Sastrawan dkk (2010)	Melakukan analisis pengaruh Combine Sampling (SMOTe+Tomek Links) dalam memprediksi Churn untuk perusahaan telekomunikasi. SMOTE + Tomek Link lebih unggul daripada metode yang lain untuk gini coefficient sedangkan untuk yang lain berada diurutan kedua.
Sevita, 2012	Menggunakan metode *bagging Logistik* pada data diagnosis kanker serviks.
Sain, 2013	Menggunakan SMOTE+Tomek Link SVM pada data medis. Hasil metode SMOTE+Tomek Link SVM, secara umum lebih baik daripada SMOTE dan Tomek Links
Trapsilasiwi, 2013	Menggunakan SMOTE LS-SVM PSO GSA untuk klasifikasi multi class pada data medis (Kanker payudara, kanker serviks). Hasil kedua percobaan belum memuaskan, masih terjadi *overfitting.*

BAB 3
METODOLOGI PENELITIAN

3.1 Sumber Data

Data yang digunakan dalam penelitian adalah data sekunder, yang diambil dari UCI *Repository Of Machine Learning*(*https://archive.ics.uci.edu/ml/machine-learning-databases/thyroid-disease/new-thyroid.data*), yang dideskripsikan pada Tabel 3.1 dan salah satu rumah sakit swasta di Surabaya, dideskripsikan pada Tabel 3.2. Studi kasus yang digunakan dalam penelitian ini merupakan data klasifikasi *multi class imbalance.* Deskripsi data yang digunakan dalam penelitian ini adalah sebagai berikut :

Tabel 3.1 Deskripsi data *Thyroid* dari UCI *Repository Of Machine Learning* Databases

No	Data	Deskripsi Data	Distribusi kelas
1	Thyroid	Data diperoleh dari UCI *Repository Of Machine Learning*, merupakan hasil dari lima lab, berdasarkan kelengkapan hasil medical, anamnesis dan *scan*yang digunakan untuk memprediksi penyakit thyroid pasien.. Jumlah prediktor (*p*) sebanyak 5 dan jumlah data (n) sebanyak 215. Data thyroid terdiri dari 3 kelas yaitu 1= Normal (*euthyroidism*) *Normal* adalah suatu keadaan dimana produksi hormon *thyroid* oleh kelenjar thyroid mencukupi dan seimbang 2=*Hypothyroidism* *Hipothyroid* adalah suatu keadaan dimana produksi hormon *thyroid* oleh kelenjar thyroid tidak mencukupi 3=*Hyperthyroidism* *Hyperthyroid* adalah suatu keadaan dimana kelenjar *thyroid* bekerja berlebihan (overactive).	1=normal (n_1=150 atau 69,76%) 2=hypothyroidism (n_2=35 atau 16,28%) 3=hyperthyroidism (n_3=30 atau 13,95%)

Tabel 3.2 Deskripsi Data *Real* dari Rumah Sakit Swasta di Surabaya

No	Data	Deskripsi Data	Distribusi Kelas
1	Kanker Payudara Breast Cancer)	Data pasien hasil biopsy. Biopsy ini dilakukan ketika test lainnya memberikan indikasi kuat bahwa seorang telah mengidap kanker payudara. Biopsy terdiri dari beberapa jenis yaitu Fine Needle Aspiration Bipsy, Core Needle Biopsy dan Open Biopsy. Data ini diambil pada Tahun 2011. Jumlah prediktor (*p*) sebanyak 6 dan jumlah data (n) sebanyak 178. Tingkat keganasan dilihat dari stadium penderita payudara yaitu : 1. Stadium 1 : peluang untuk hidup dalam waktu 5 tahun sebesar 87% 2. Stadium II : peluang untuk hidup dalam waktu 5 tahun sebesar 75% 3. Stadium III : peluang untuk hidup dalam waktu 5 tahun sebesar 46%	1= "Stadium 1 " (n_1=11 atau 6%) 2= "Stadium II" (n_2=67 atau 38%) 3= "Stadium III " (n_3=100 atau 56%)
2	Kanker Serviks (Cervical Cancer)	Data pasien dari hasil *pap smear* pada Tahun 2010. *Pap Smear* merupakan tes skrining untuk mendeteksi dini perubahan atau abnormalitas dalam serviks sebelum sel-sel tersebut menjadi kanker. Jumlah prediktor (*p*) sebanyak 7 dan jumlah data (n) sebanyak 794. Klasifikasi *Pap Smear* menurut *Papanicolaou* yaitu : 1. Kelas I : normal smear 2. Kelas II : menunjukan adanya infeksi ringan non spesifik, terkadang disertasi dengan kuman atau virus tertentu dan disertai pula dengan kariotik ringan. 3. Kelas III: ditemukan sel diagnostik dengan keradangan berat 4. Kelas IV : ditemukan sel-sel yang mencurigakan ganas 5. Kelas V : ditemukan sel-sel ganas	1= "Kelas I" (n_1=299 atau 38%) 2= "Kelas II" (n_2=340 atau 43%) 3= "Kelas III " (n_3=98 atau 12%) 4= "Kelas IV " (n_4=50 atau 6%) 5= "Kelas V " (n_5=7 atau 1%)

3.2 Variabel Penelitian

Variabel yang digunakan dalam penelitian ini adalah sebagai berikut :

Tabel 3.3 Variabel Data Kanker Payudara (*Breast Cancer*)

Simbol	Variabel	Keterangan	Skala
X_1	Ukuran Tumor	0=”teraba tumor dengan diamter kurang dari 2 cm” 1=”teraba tumor dengan diameter antara 2 sampai dengan 5 cm” 2=”teraba tumor dengan diamter lebih dari 5 cm” 3=”teraba tumor dengan diamter sangat besar”	Ordinal
X_2	Nodus Salah satu komponen dari sistem limfatik yang dapat ditemukan pada tubuh manusia. Nodus limfa adalh filter untuk filter untuk partikel asing dan berisi sel darah putih	0=”tidak ada metastase (penyebaran sel kanker) regional” 1=”ada metastase kelenjar aksila yang mobile” 2=”ada metastase kelenjar aksila yang melekat” 3=”metastase kelenjar mummae internal”	Ordinal
X_3	Kemoterapi Merupakan program penggabungan beberapa preparat untuk meningkatkan penghancuran sel tumor dan untuk meminimalkan resistensi medikal	0=”melakukan kemoterapi” 1=”tidak melakukan kemoterapi”	Nominal
X_4	Tingkat keganasan	0= “ganas” 1= “jinak”	Nominal
X_5	Letak kanker	0= “kiri” 1=”kanan”	Nominal
X_6	Usia pasien	0=”23-41 tahun” 1=”42-60 tahun” 2=”61-79tahun”	Nominal
Y	Jenis Stadium Pasien	1= “Stadium I” 2= “Stadium II” 3= “Stadium III”	Ordinal

(Trapsilasiwi, 2013; Rahman, 2012)

T**abel 3.4** Variabel Data Kanker Serviks (*Cervical Cancer*)

Simbol	Variabel	Keterangan	Skala
X_1	Usia pasien saat melakukan pemeriksaan		Rasio
X_2	Penggunaan Kontrasepsi	1="tidak menggunakan alat kontrasepsi" 2="menggunakan alat kontrasepsi"	Nominal
X_3	Usia menstruasi pertama kali		Rasio
X_4	Usia pertama kali melahirkan		Rasio
X_5	Paritas yaitu jumlah anak yang pernah dilahirkan baik hidup maupun sudah meninggal	1= "paritas ≤ 2 orang" 2="paritas > 2 orang"	Nominal
X_6	Siklus menstruasi	1=" teratur" 2=" tidak teratur"	Nominal
X_7	Riwayat keguguran	1= tidak pernah keguguran 2 = pernah keguguran	Nominal
Y	Hasil *Pap Test*	1= "Kelas 1" Jumlah data 29 atau 38% 2= " Kelas 2" Jumlah data 340 atau 43% 3="Kelas 3" Jumlah data 98 atau 12% 4= " Kelas 4" Jumlah data 50 atau 6% 5= "Kelas 5" Jumlah data 7 atau 1%	Ordinal

(Trapsilasiwi, 2013; Sevita, 2012).

Tabel 3.5 Variabel Data *Thyroid*

Simbol	Variabel	Keterangan	Skala
X_1	Persentase hasil uji asam T3 (T3 resin)		Rasio
X_2	Total Serum Thyroxin (T4)	Diukur oleh metode *isotopic dispalcement*	Rasio
X_3	Total serum *triidothyronine*	Diukur *oleh radioimmuno assay*	Rasio
X_4	*Hormon basal thyroid stimulating* (TSH)	Diukur *oleh radioimmuno assay*	Rasio
X_5	Perbedaan *maximal absolute* pada nilai TSH setelahdisuntik		Rasio
Y	Kondisi Thyroid	1= Normal 2= "Hyperthyroidism" 3= "Hypothyroidism"	Ordinal

3.3 Metode Penelitian

Adapun metode penelitian yang dilakukan pada penelitian ini terdiri dari :

1. Mendesain algoritma *Combine Sampling (SMOTE+Tomek Links)* dengan langkah sebagai berikut
   A. Melakukan penanganan masalah kondisi im*balanced* dengan menggunakan algoritma SMOTE.
   B. Melakukan penanganan masalah kondisi im*balanced* dengan menggunakan algoritma Tomek Links.
   C. Melakukan penanganan masalah kondisi im*balanced* dengan menggunakan *Combine Sampling*, yaitu dengan penggunaan metode SMOTE kemudian dilanjutkan ke Tomek Links
2. Menerapkan metode *Combine Sampling (SMOTE+Tomek Links)* LS-SVM, dengan langkah-langkah sebagai berikut :
   A. Melakukan *Preprocessing* data
   B. Melakukan Deskripsi Data
   C. Melakukan *Preprocessing imbalanced* data (SMOTE, Tomek Links dan Combine Sampling)
   D. Melakukan klasifikasi LS-SVM OAO untuk kasus klasifikasi *multi class*

i. Membagi data menjadi data *training* dan *testing* dengan menggunakan 5 dan 10 *fold crossvalidation*.

ii. Menentukan nilai parameter σ dan C (C=1,50,100 dan σ=1,10, 20)

E. Mengoptimisasi parameter kernel RBF (*Radial Basis Function*) dan nilai *C* (nilai pinalti) pada LS-SVM dengan PSO-GSA sehingga tidak melakukan *trial and error* untuk penentuan parameternya.

Flowchart Combine LS-SVM PSO-GSA dapat dilihat pada Gambar 3.1

F. Mengevaluasi performansi metode klasifikasi *Combine Sampling* LS-SVM PSO-GSA berdasarkan nilai akurasi total, *Sensitivity, Specificity, Precision, Fmeasure dan Gmean.*

G. Melakukan perbandingan perfomansi metode pada setiap data berdasarkan nilai akurasi, *Sensitivity* dan *G-mean* dengan menggunakan Uji *Friedman*. Metode yang diukur performansinya antara lain :

M1= LS-SVM

M2= SMOTE LS-SVM

M3= Tomek Links LS-SVM

M4= Combine LS-SVM

M5= LS-SVM PSO-GSA

M6= SMOTE LS-SVM PSO-GSA

M7= Tomek Links LS-SVM PSO-GSA

M8= Combine LS-SVM PSO-GSA

Struktur data untuk uji Friedman dapat dilihat pada Tabel 3.6.

Tabel 3.7 Struktur Data Perbandingan Metode Klasifikasi dengan Uji Friedman

Data (i) (blok)	Metode (rata-rata Akurasi)/ Perlakuan (j)							
	M1	M2	M3	M4	M5	M6	M7	M8
Thyroid (1)	$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,6}$	$x_{1,7}$	$x_{1,8}$	$x_{1,9}$
Kanker Payudara (2)	$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$x_{2,6}$	$x_{2,7}$	$x_{2,8}$
Kanker Serviks (3)	$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$	$x_{3,5}$	$x_{3,6}$	$x_{3,7}$	$x_{3,8}$
Jumlah peringkat	R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8

Dimana

x_{ij} = rata-rata akurasi/*sensitivity*/*specificity* pada data ke-*i* metode ke-*j* $(j = 1, 2, ..., 8); (i = 1, 2, 3)$

R_j = jumlah peringkat pada perlakuan ke -*j* $(j = 1, 2, ..., 8)$

Jika dalam pengujian hipotesis diputuskan Tolak H_0 maka dilanjutkan ke uji perbandingan berganda

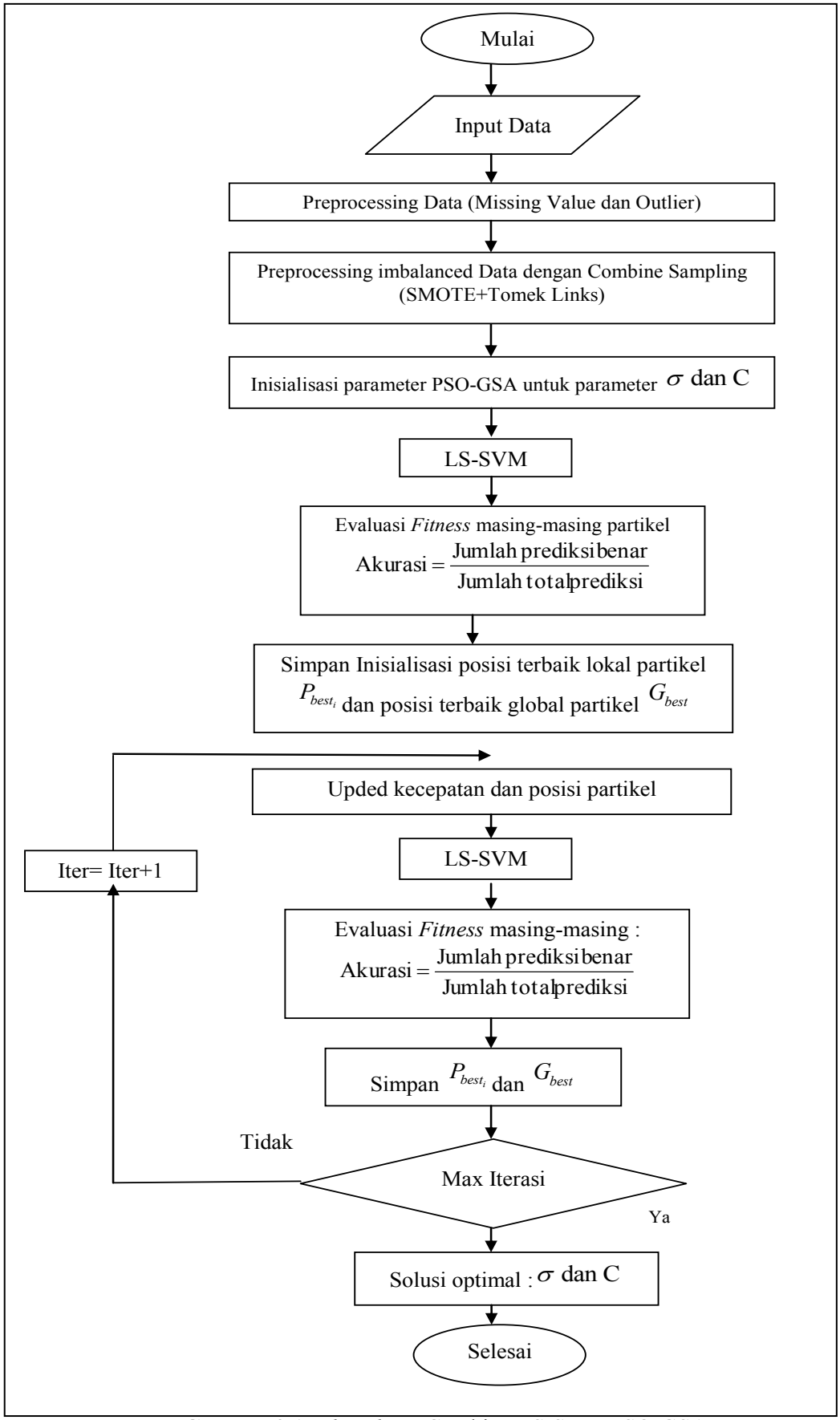
H. Melakukan perbandingan *Cross validation* dengan uji *Mann whitney.*

Dilakukan uji dua sampel independen *Mann whitney* pada setiap data. Sampel dari fold 5 dan fold 10 adalah independen. Struktur data untuk uji Mann Whitney dapat dilihat pada Tabel 3.7.

Tabel 3.7 Struktur Data Perbandingan *Cross Validation* dengan Uji Mann Whitney

Metode	Akurasi 5 Fold	Akurasi 10 Fold	*Sensitivity* 5 Fold	*Sensitivity* 10 Fold	*G-mean* 5 Fold	*G-mean* 10 Fold
M1	$x_{1,1}$	$x_{1,2}$	$x_{1,1}$	$x_{1,2}$	$x_{1,1}$	$x_{1,2}$
M2	$x_{2,1}$	$x_{2,2}$	$x_{2,1}$	$x_{2,2}$	$x_{2,1}$	$x_{2,2}$
⋮	⋮	⋮	⋮	⋮	⋮	⋮
M8	$x_{8,1}$	$x_{8,2}$	$x_{8,1}$	$x_{8,2}$	$x_{8,1}$	$x_{8,2}$
Jumlah rangking	R_1	R_2	R_1	R_2	R_1	R_2

3. Kesimpulan



Gambar 3.1 *Flowchart* Combine LS-SVM PSO-GSA

BAB 4
HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan tentang desain algoritma Combine LS-SVM PSO-GSA dan penerapan metode Combine Least Square Support Vector pada data medis kemudian membandingkan performansi metode dengan menggunakan uji *Friedman* dan membandingkan q-*fold cross validation* terbaik dengan pengujian *Mann Whitney*.

4.1 Desain Algoritma Combine Sampling Least Square Support Vector Machine PSO-GSA.

Pada penelitian ini menggunakan metode Combine Least Square Support Vector Machine dan optimasi parameter PSO-GSA. Berikut ini merupakan algoritma Combine LS-SVM PSO-GSA.

Algoritma 1. Combine Sampling

Algoritma 1.1 SMOTE

Input : Jumlah data minoritas kelas (T); Jumlah data mayoritas kelas (P) Jumlah replikasi SMOTE (N) ; Jumlah *nearest neighbor* (*knn)*

Output: data sintetis x_{syn}

Begin :

1. Menentukan banyaknya data dari kelas minor (T), dikatakan kelas minor jika persentase jumlah data kelas kurang dari 50%.
2. Menentukan banyaknya data dari kelas mayor, hanya ada 1 kelas data mayor.
3. Menghitung k *nearest neighbor* atau jarak eulidean pada semua data

   for x=1:n

   for z=1:n

   $$d(\mathbf{x},\mathbf{z}) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2 + \ldots + (x_p - z_p)^2};$$

   distance = d(**x**,**z**)

   end;

3. Menentukan replikasi untuk data minoritas $N=\frac{\text{Jumlah data kelas mayor (P)}}{\text{Jumlah data kelas minor (T)}}$

4 Menentukan data yang akan direplikasi pada kelas data minor x_i

5 Menentukan data dengan jarak terdekat dari data yang akan direplikasi dalam satu kelas minor yang sama (x_{knn}).

6. Menentukan nilai randoman γ (γ bilangan randoman antara nilai [0,1])

6 Menghitung sintetis dengan menggunakan persamaan (2.1)

$$x_{syn} = x_i + (x_{knn} - x_i) \times \gamma$$

Algoritma 1.2 Tomek Links

Input : data kelas mayor (**x**); data kelas minor (**z**)

Output: data Eliminasi kelas mayor yang merupakan kasus Tomek Links

Begin :

1. Mengidentifikasi kelas negative/mayor (**x**) dan data kelas positif/minor (**z**)
2. Menghitung Jarak Eulidean $d(\mathbf{x},\mathbf{z}) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2 + \ldots + (x_p - z_p)^2}$

```
for x=1:n
  for z=1:n
      d(x,z) = sqrt((x1 - z1)^2 + (x2 - z2)^2 + ... + (xp - zp)^2);
  distance = d(x,z)
    end;
```

3. Mengidentifikasi data kelas mayor dan minor (z*) yang dekat dengan data kelas mayor dan minor.

```
For k=1 : n
        If distance (k)=min(distance)
                Z* = k
end
```

4. Menghitung jarak eulidean $d(\mathbf{x},\mathbf{z}^*) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2 + \ldots + (x_p - z_p)^p}$

```
for x=1:n
  for z=1:n
```

for z*=1:n

$$d(\mathbf{x},\mathbf{z}^*) = \sqrt{(x_1 - z_1^*)^2 + (x_2 - z_2^*)^2 + \ldots + (x_p - z_p^*)^2}\ ;$$

$$d(\mathbf{x},\mathbf{z}) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2 + \ldots + (x_p - z_p)^2}\ ;$$

end;

5. Deteksi kasus Tomek Links

   Sepasang (x,z) disebut Tomek Links jika tidak ada sampel **z***, sehingga $d(\mathbf{x},\mathbf{z}^*) < d(\mathbf{x},\mathbf{z})$ atau $d(\mathbf{z},\mathbf{z}^*) < d(\mathbf{z},\mathbf{x})$

4 Jika terdeteksi Tomek Links maka data mayor (x) dihapus

Algoritma 1.3 Combine Sampling

Input : data kelas mayor baru hasil SMOTE (**x**); data kelas minor baru hasil SMOTE (**z**)

Output: data Eliminasi kelas mayor yang merupakan kasus Tomek Links

Begin :

1. Mengidentifikasi kelas negative/mayor (**x**) dan data kelas positif/minor (**z**)
2. Menghitung Jarak Eulidean $d(\mathbf{x},\mathbf{z}) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2 + \ldots + (x_p - z_p)^2}$

   for x=1:n

   for z=1:n

$$d(\mathbf{x},\mathbf{z}) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2 + \ldots + (x_p - z_p)^2}\ ;$$

   distance = d(**x**,**z**)

   end;
3. Mengidentifikasi data kelas mayor dan minor (z*) yang dekat dengan data kelas mayor dan minor.

   For k=1 : n

   If distance (k)=min(distance)

   Z* = k

   end
4. Menghitung jarak eulidean $d(\mathbf{x},\mathbf{z}^*) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2 + \ldots + (x_p - z_p)^p}$

for x=1:n

for z=1:n

for z*=1:n

$$d(\mathbf{x},\mathbf{z}^*) = \sqrt{(x_1 - z_1^*)^2 + (x_2 - z_2^*)^2 + \ldots + (x_p - z_p^*)^2}\ ;$$

$$d(\mathbf{x},\mathbf{z}) = \sqrt{(x_1 - z_1)^2 + (x_2 - z_2)^2 + \ldots + (x_p - z_p)^2}\ ;$$

end;

5. Deteksi kasus Tomek Links

   Sepasang (x,z) disebut Tomek Links jika tidak ada sampel **z***, sehingga $d(\mathbf{x},\mathbf{z}^*) < d(\mathbf{x},\mathbf{z})$ atau $d(\mathbf{z},\mathbf{z}^*) < d(\mathbf{z},\mathbf{x})$

4 Jika terdeteksi Tomek Links maka data mayor (x) dihapus

Algoritma 2. Least Square Support Vector Machine OAO PSO-GSA

Input : Data input (X), data target (Y), parameter kernel (σ) , penalti (C)

Output: α,b, akurasi

Begin :

Tahap Training :

1. Membagi kelas menjadi biner *v(v − 1) /2*

   For *k*= 1: *v*, l = *k*+1 : *v*

   Jika kelas *k* di*taining* dengan kelas *l*
2. Menentukan parameter fungsi kernel. Pada penelitian ini menggunakan fungsi kernel RBF.
3. Menghitung matriks kernel RBF (*K)*
4. Menentukan fungsi Tujuan (*objective*) LS-SVM, sesuai persamaan (2.22)
5. Menentukan parameter penalti C
6. Meminimumkan fungsi *langrange primal*, sesuai persamaan (2.24)
7. Merubah bentuk ke sistem linier, sesuai persamaan (2.25) sebagai pengganti *Quadratic Programming*
8. Hitung nilai (α, b) dengan Optimasi dengan Karush Kuhn Tucker (KKT)
9. Membentuk persamaan *construct hyperplane* : $\hat{f}^{kl}(\mathbf{x}) = sign(\hat{\mathbf{w}}^{kl^T}\mathbf{x}) + b^{kl} = 0$

Tahap Testing :

1. Initial *voting* pada setiap kelas
2. Jika data **x** dimasukkan dalam ke dalam persamaan *construct* dan hasilnya menyatakan x adalah kelas k

   Kemudian

   Voting (*k*)= voting (*k*) +1

   Else

   Voting (*l*) = voting (*l*)+1

   End if

   Kelas dari x ditentukan dari jumlah *voting* terbanyak.

Tahap PSO-GSA

1. Inisialisasi jumlah partikel (*swarm*), maksimum iterasi, bobot maksimum, bobot minimum dan β dan G_0 untuk GSA dan parameter σ dan C
2. Menentukan posisi partikel $\theta(C,\sigma)_i^p$ dan kecepatan $\boldsymbol{V}_i^p(t)$ awal
3. Melakukan perhitungan fungsi obyektif untuk mendapatkan *fitness*

   $$\text{Akurasi Total} = \frac{\text{Jumlah prediksi benar}}{\text{Jumlah total prediksi}}$$

4. Menghitung nilai gravitasi $G(t) = G_0 \times \exp(-\beta \times \frac{t^*}{T^*})$
5. Menghitung nilai percepatan, dimana digunakan untuk menggantikan *Pbest* pada PSO $ac_i^p(t) = \frac{F_i^p(t)}{M_i(t)}$
6. Dari perhitungan fungsi obyektif didapatkan nilai *Gbest* (*Global best position*)
7. Memperbaruhi kecepatan *(upded velocity)* dan posisi *(upded position)*

   Kecepatan *(upded velocity)* :

   $$\boldsymbol{V}_i^p(t+1) = \omega \times \boldsymbol{V}_i^p(t) + c_1 r \times ac_i^p(t) + c_2 r \times \left(\boldsymbol{Gbest}^p(t) - \theta_i^p(t)\right)$$

   Posisi *(upded position)*

   $$\theta_i^p(t+1) = \theta_i^p(t) + \boldsymbol{V}_i^p(t+1)$$

8. Mengulangi langkah nomor 1 sampai 8 hingga memenuhi iterasi yang telah ditentukan.

4.2 Penerapan Combine Least Square Support Vector Machine

4.2.1 Preprocessing Data

Tahap *preprocessing* data merupakan tahap awal sebelum data siap diolah. *Preprocessing* data meliputi deteksi *missing value* dan deteksi *outlier.* Berikut ini merupakan uraian tentang deteksi *missing value* dan deteksi outlier pada data kanker payudara, kanker serviks dan data thyroid.

a. Data hasil biopsy kanker payudara

Data dari hasil *biopsy* dari rumah sakit sebanyak 500. Dari 500 data, dilakukan deteksi *missing value*. Data yang *missing value* dihapus karena lebih dari 30%. Jumlah data keseluruhan yang lengkap berjumlah 178.

b. Data hasil *pap smear* kanker serviks

Dari 794 data, ditemukan *missing value* pada variabel usia pasien saat pertama kali menstruasi. *Missing value* pada variabel usia pasien diisi dengan rata-ratanya.

c. Data Thyroid

Pada data thyroid tidak ditemukan data yang *missing* value tetapi terdapat data yang outlier. Pengujian outlier secara *multivariate* adalah sebagai berikut.

Hipotesis :

H_0 : Data ke-*i* tidak outlier

H_1 : Data ke-*i* outlier

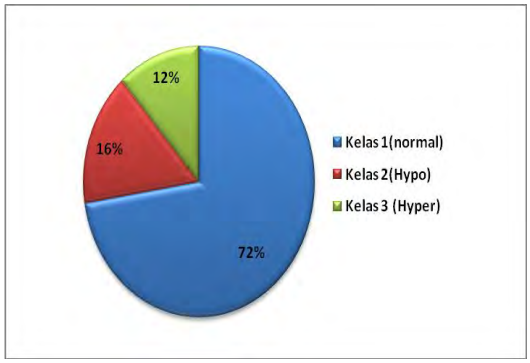
Hasil pengujian outlier secara *multivariate*, diperoleh nilai *P-Value* kurang dari tingkat signifikan (0,00001) maka disimpulkan terdapat *outlier* pada observasi ke-156, 167, 193, 195, 196,199,208 dan 204. Observasi ini dihilangkan. Data thyroid menjadi 207. Hasil pengujian *outlier* secara multivariate dapat dilihat pada Lampiran 1.

4.2.2 Deskripsi Data

Setelah data terbebas dari *missing value* dan *oulier* maka selanjutnya akan dilakukan deskripsi data pada ketiga kasus yaitu kasus Thyroid, kanker payudara dan kanker serviks dijelaskan pada sub bab ini. Setiap studi kasus memiliki karakteristik yang berbeda.

4.2.2.1 Thyroid

Deskripsi data mengenail persentase masing-masing kelas untuk kondisi pasien thyroid ditunjukkan pada Gambar 4.1.



Gambar 4.1 Kondisi Pasien Thyroid

Gambar 4.1 menunjukkan bahwa jumlah pasien yang menderita thyroid dengan kondisi normal paling tinggi yaitu sebesar 72% sedangkan pasien dengan kondisi *hypothyroid* dan *hyperthyroid* sebesar 16% dan 12%.

Deskripsi data untuk masing-masing variabel indicator dapat dilihat pada Tabel 4.1.

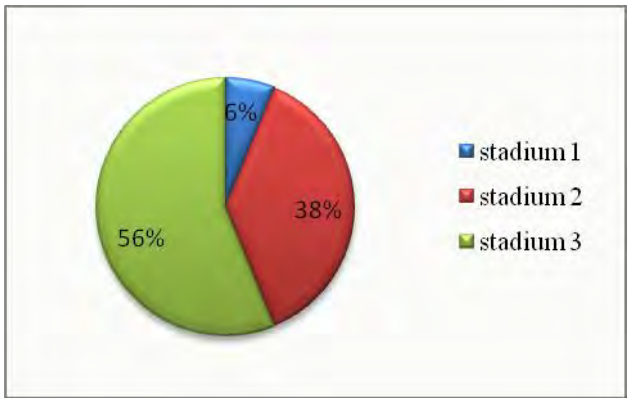
Tabel 4.1 Deskripsi Data Thyroid

Variabel	Rata-rata	Standart Deviasi	Minimum	Maksimum
Persentase uji T3 Resin (X1)	41,359	12,65	65,00	144,00
Total Serum Thyroxin (T4) (X2)	12,929	4,510	0,500	25,30
Total Serum Triidothyronine (X3)	25,780	1,199	0,200	7,80
Hormon TSH (X4)	2,176	3,440	0,100	23,00
Perbedaan absolute TSH (X5)	3,361	5,456	-0,700	40.80

Tabel 4.1 dapat diketahui bahwa rata-rata persentase uji T3 resin sebesar 41,359, nilai minimum 65 dan maksimum 144. Rata-rata Total Serum *Thyroxin* sebesar 12,929, nilai minimum 0,5 dan maksimum 25,30. Rata-rata Total Serum *Triidothyronine* sebesar 25,780, nilai minimum 0,2 dan maksimum 7,80. Rata-rata Perbedaan absolute TSH sebesar 3,361, nilai minimum -0,7 dan maksimum 40,80. Pasien Thyroid cenderung menghasilkan kadar persentase uji T3 Resin yang tinggi.

4.2.2.2 Kanker Payudara

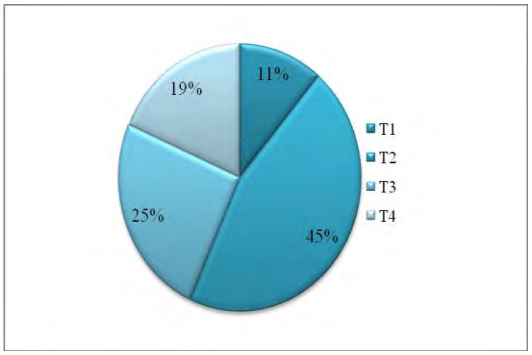
Deskripsi data mengenai persentase masing-masing kelas untuk pasien kanker payudara ditunjukkan pada Gambar 4.2.



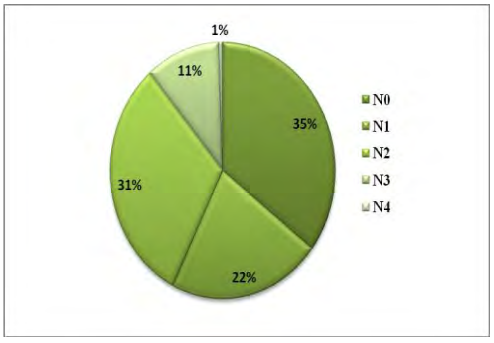
Gambar 4.2 Jenis Stadium Pasien

Dari Gambar 4.2 dapat dilihat bahwa jumlah pasien yang menderita kanker payudara Stadium III paling tinggi yaitu sebesar 56%, sedangkan pasien yang tergolong dalam Stadium I dan II sebesar 6% dan 38%.

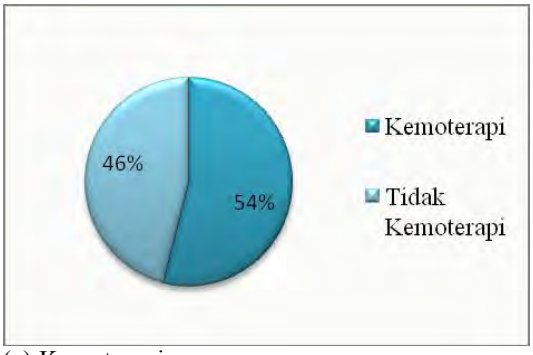
Deskripsi untuk masing-masing variabel indikator yaitu usia, ukuran tumor, nodus, kemoterapi, tingkat keganasan, dan letak kanker dipaparkan sebagaimana berikut ini.



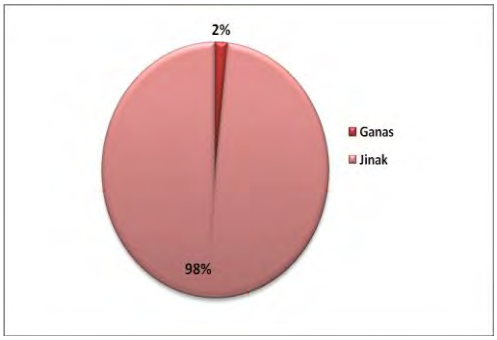
(a) Ukuran Tumor



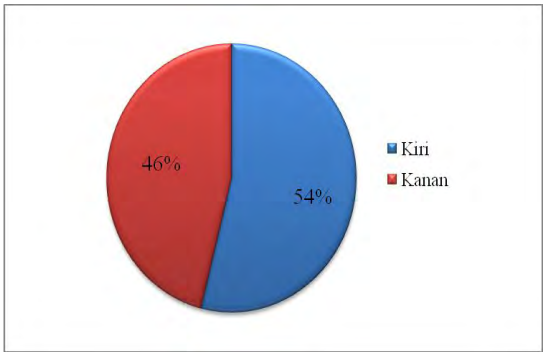
(b)Tipe Nodus



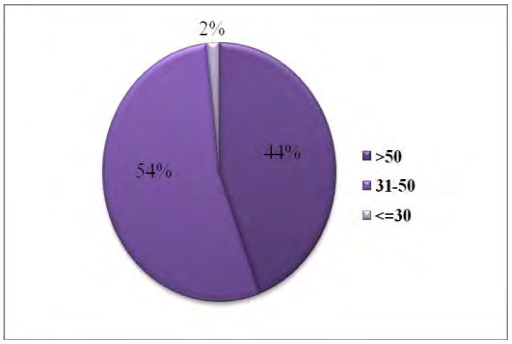
(c) Kemoterapi



(d) Tingkat Keganasan



(e) Letak Kanker



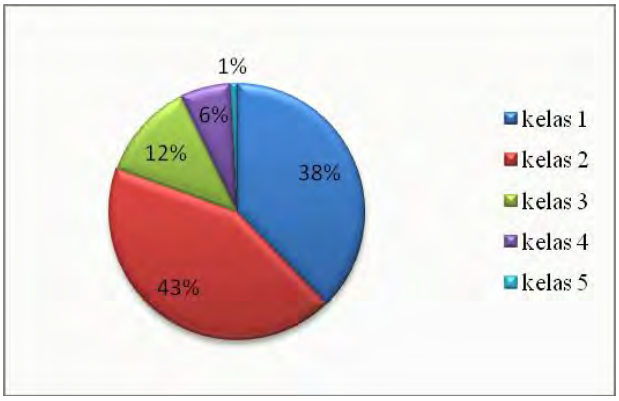
(f) Usia Pasien

Gambar 4.3 Faktor Penyebab Kanker Payudara (a) ukuran Tumor; (b) Tipe Nodus; (c) Kemoterapi; (d) Tingkat Keganasan; (e) Letak Kanker; (f) usia pasien

Gambar 4.3 menunjukkan bahwa mayoritas ukuran tumor pasien berdiameter sangat besar yaitu sebesar 45%. Persentase pasien yang tidak mengalami penyebaran sel-sel kanker sebesar 35%. Persentase pasien yang melakukan kemoterapi sebesar 54% sedangkan persentase pasien yang tidak melakukan kemoterapi sebesar 46%. Pasien mayoritas menderita kanker ringan (jinak) yaitu sebesar 98%. Letak kanker pasien cenderung berada disebelah kiri. Mayoritas umur pasien kanker payudara yaitu 31-50 tahun.

4.2.2.3 Kanker Serviks

Pada Gambar 4.4 telah dipaparkan statistika deskriptif untuk klasifikasi hasil *pap smear* para pasien kanker serviks.



Gambar 4.4 Hasil Pap Smear Pasien Kanker Serviks

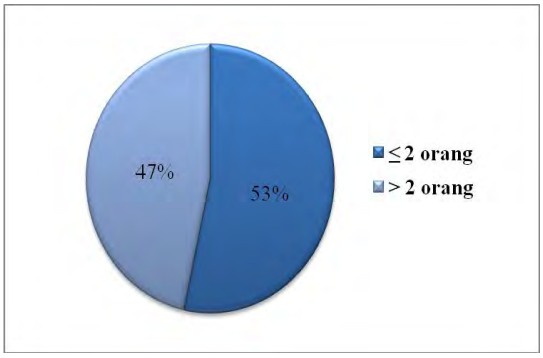
Gambar 4.4, diketahui bahwa masing-masing persentase kelas 1, 2, 3, 4, dan 5 adalah sebesar 38%, 43%, 12%, 6%, dan 1%. Kelas 2 memiliki persentase tertinggi, hal ini berarti sebagian besar pasien kanker serviks didiagnosa mengalami radang ringan non spesifik dan terdapat sel-sel abnormal sedangkan persentase terendah dimiliki oleh kelas 5, yang artinya adalah sedikit sekali pasien kanker serviks yang didiagnosa terdapat sel-sel ganas pada serviksnya.

Statistika deskriptif yang dapat memberikan gambaran mengenai masing-masing variabel indikator dipaparkan sebagaimana berikut.

Tabel 4.2 Deskripsi Data Usia Pasien Kanker Serviks

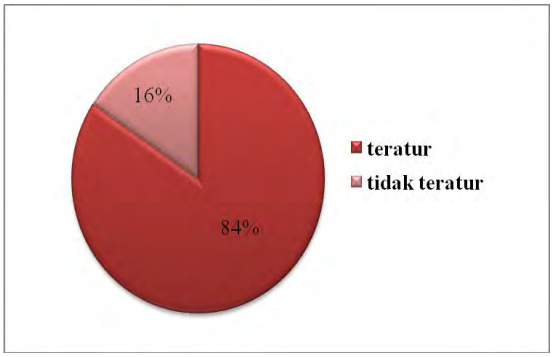
Variabel	Rata-rata	Minimum	Maksimum
Usia saat pemeriksaan	41	21	78
Usia pertama kali menstruasi	13	9	19
Usia pertama kali melahirkan	26	12	51

Berdasarkan Tabel 4.2 dapat diketahui bahwa rata-rata pasien yang melakukan pemeriksaan kanker serviks adalah berusia 41 tahun. Pasien termuda yang melakukan pemeriksaan berusia 21 tahun sedangkan pasien tertua berusia 78 tahun. Berdasarkan rata-rata, usia pasien yang mengalami menstruasi pertama kali berusia 13 tahun, dengan pasien termuda berusia 9 tahun dan pasien tertua berusia 19 tahun. Usia maksimum saat pertama kali melahirkan adalah pasien berusia 51 tahun sedangkan usia minimum adalah pasien berusia 12 tahun, rata-rata pasien melahirkan pertama kali saat berusia 26 tahun.

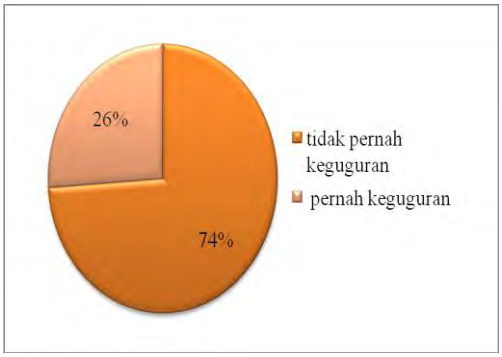


(a) Paritas

(b)Penggunaan Alat Kontrasepsi



(c) Siklus Menstruasi



(d) Riwayat Keguguran

Gambar 4.5 Faktor Penyebab Kanker Serviks (a) Paritas; (b) Penggunaan Alat Kontrasepsi; (c) Siklus Menstruasi; (d) Riwayat Keguguran

Berdasarkan Gambar 4.5 diketahui bahwa Persentase pasien yang pernah melahirkan anak ≤ 2 orang sebesar 53% sedangkan prosentase pasien yang pernah melahirkan anak > 2 orang sebesar 47%. Persentase pasien yang tidak menggunakan kontrasepsi sebesar 55%. Persentase pasien yang siklus menstruasinya teratur sebesar 84% sedangkan persentase pasien yang siklus menstruasinya tidak teratur sebesar 16%. Persentase pasien yang tidak pernah mengalami keguguran sebesar 74% sedangkan persentase pasien yang pernah mengalami keguguran sebesar 26%.

4.2.3 Preprocessing Data *Imbalanced Data* Pada Kasus Klasifikasi *Multiclass*

Pada tahap *preprocessing* data *imbalanced* dilakukan untuk mempersiapkan data asli sebelum data tersebut siap diolah menggunakan metode yang digunakan dalam penelitian ini. Data yang digunakan dalam penelitian ini harus merupakan kasus *multiclass*, yaitu variabel respon memiliki lebih dari dua kategori. Preprocessing data Imbalanced pada penelitian menggunakan SMOTE, Tomek Links dan *Combine Sampling*.

4.2.3.1 Metode SMOTE

Metode SMOTE merupakan metode *oversampling* yang digunakan untuk meningkatkan jumlah kelas minoritas dengan mereplikasi data secara acak sesuai dengan persentase yang diinginkan sehingga jumlahnya mendekati jumlah data mayor. Penerapan metode *oversampling* pada data imbalanced menyebabkan tingkat imbalanced data semakin kecil dan klasifikasi dapat dilakukan dengan tepat. Hasil dari penanganan metode SMOTE terhadap masing-masing data *imbalanced* yang digunakan dalam dalam penelitian ini ditampilkan dalam Tabel 4.3 dan Gambar 46.

Tabel 4.3 Deskripsi Distribusi Data Sebelum dan Setelah SMOTE

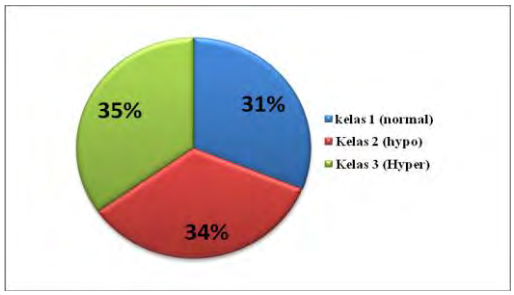
Data	Kelas Mayor	Kelas Minor	Replikasi	Kelas Mayor Baru	Kelas Minor Baru
Thyroid	(150*)(72 %**) (1***)	(33*)(16%**) (2***)	4 kali	(150*)(31%**) (1***)	(165*)(34%**) (2***)
		(24*)(12%**) (3***)	6 kali		**(168*)(35%**) (3***)**
Kanker Payudara	(100*) (56%**) (3***)	(11*)(6%**) (1***)	9 kali	(100*)(29%**) (3***)	(110*) (32%**) (1***)
		(67*)(38%**) (2***)	1 kali		**134 (39%) (2)**
Kanker Serviks	(340*) (43%**) (2***)	(7*) (1%**) (5***)	6 kali	(340*)(20%**) (2***)	#**49**
		#**49**	6 kali		(343*) (20%**) (5***)
		(50*) (6%**) (4***)	6 kali		(350*) (20%**) (4***)
		(98*)(12%**) (3***)	3 kali		**(392*)(23%**) (3***)**
		(299*)(38%**) (1***)	-		(299*) (17%**) (1***)

Keterangan: #) angka yang digunakan adalah sama

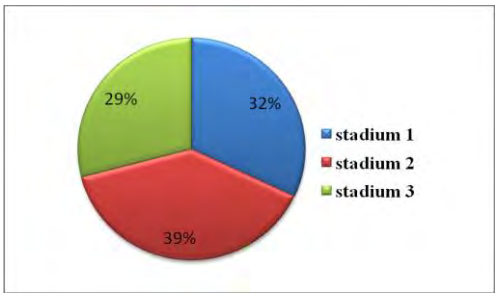
*) jumlah data , **)persentase data, ***) kategori kelas

Tabel 4.3, telah dipaparkan deskripsi distribusi data thyroid, kanker payudara dna kanker serviks. Pada thyroid, kelas dengan jumlah anggota sedikit adalah kelas 2 (hypothyroid) dan kelas 3 (hyperthyroid). Setiap data pada kelas 2 dan kelas 3 akan direplikasi sehingga jumlah data akan meningkat dan menyeimbangi jumlah data pada kelas mayor. Pada kanker payudara, kelas dengan jumlah anggota sedikit adalah kelas 1 dan kelas 2. Setiap data pada kelas 1 dan kelas 2 akan direplikasi sehingga jumlah data akan meningkat dan menyeimbangi jumlah data pada kelas mayor. Pada data kanker serviks, anggota kelas 5 hanya berjumlah 7 data sehingga dilakukan replikasi sebanyak 2 tahap. Hal ini dikarenakan jumlah maksimum tetangga terdekat (*nearest neighbor*) adalah 10 artinya replikasi maksimum yang dapat dilakukan sebanyak 11 kali. Apabila dilakukan replikasi 11 kali pada data tersebut maka jumlah data baru yaitu 84 data sedangkan jumlah data mayor sebanyak 340. Keaadaan ini masih memiliki selisih yang sangat jauh dengan jumlah data mayor. Oleh karena itu, dilakukan replikasi 2 tahap untuk kasus seperti ini. Berdasarkan kasus ini dapat diambil kesimpulan bahwa replikasi pada jumlah data yang sangat sedikit dapat dilakukan dengan strategi replikasi dengan beberapa kali tahapan. Berkebalikan dengan kasus kelas 5, kasus kelas 1 yang memiliki jumlah anggota 299 tidak dilakukan replikasi. Penyebabnya adalah apabila dilakukan replikasi, jumlah data baru akan jauh melampaui jumlah data pada kelas mayor.

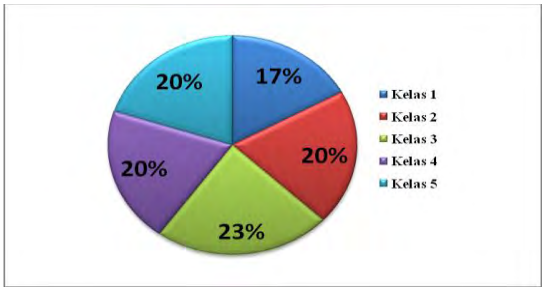
Ilustrasi distribusi data setelah dilakukan SMOTE dapat dilihat pada Gambar 4.6.



(a) Distribusi Kelas Thyroid Setelah SMOTE



(b) Distribusi Kelas Kanker Payudara



(c) Distribusi Kelas Kanker Serviks

Gambar 4.6 Distribusi Kelas Setelah SMOTE (a) Data Thyroid; (b) Data Kanker Payudara (c) Data Kanker Serviks

Pada data thyroid setelah dilakukan SMOTE yaitu pasien dengan kondisi thyroid normal sebesar 31%, kondisi pasien hypothyroid sebesar 34% dan kondisi pasien hyperthyroid sebesar 35%, diilustrasikan pada Gambar 4.6(a). Pada data kanker payudara setelah dilakukan SMOTE yaitu pasien dengan stadium I sebesar 32%, pasien dengan stadium II sebesar 39%, pasien dengan stadium III sebesar 29%, diilustrasikan pada Gambar 4.6(b). Pada data kanker serviks setelah dilakukan SMOTE yaitu pasien dengan kondisi kelas I sebesar 17%, pasien dengan kondisi kelas 2 sebesar 20%, kondisi kelas 3 sebesar 23%, kondisi kelas 4 sebesar 20% dan kondisi kelas 5 sebesar 20%.

4.2.3.2 Metode Tomek Links

Metode Tomek Links yang digunakan dalam penelitian ini yaitu metode tomek links yang dapat digunakan sebagai metode undersampling yaitu hanya kelas mayoritas yang akan dieliminasi. Penerapan metode tomek links menggunakan data asli. Hasil dari penanganan dari metode ini terhadap masing-masing data *imbalanced* yang digunakan dalam penelitian ini diilustrasikan pada Gambar 4.7. Deskripsi distribusi data menggunakan tomek links dapat dilihat pada Tabel 4.4.

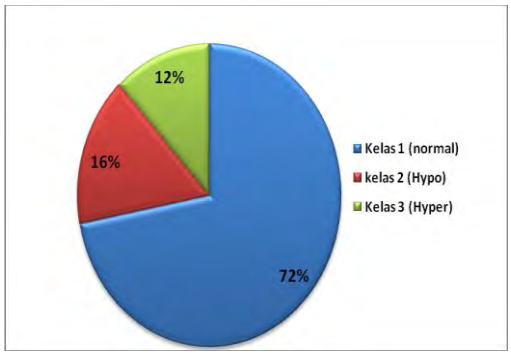
Tabel 4.4 Deskripsi Distribusi Data Sebelum dan Setelah Tomek Links

Data	Mayor	Minor	Data Mayor Hapus	Mayor Baru	Minor Baru
Thyroid	(150*)(72 %**) (1***)	(33*)(16%**) (2***) (24*)(12%) (3)	3	(147*)(72%**) (1***)	(33*)(16%**) (2***) (24*)(12%**) (3***)
Kanker Payudara	(100*) (56%**) (3***)	(11*) (6%**) (1***) (67*)(38%**) (2***)	24	(76*)(49%**) (3***)	(11*)(7%**) (1***) (67*)(44%**) (2***)
Kanker Serviks	(340*) (43%**) (2***)	(7*) (1%**) (5***) (50*) (6%**) (4***) (98*)(12%**) (3***) (299*)(38%**) (1***)	176 -	(164*)(27%**) (2***)	(7*)(1%**) (5***) (50*)(8%**) (4***) (98*)(16%**) (3***) (299*)(48%*) (1***)

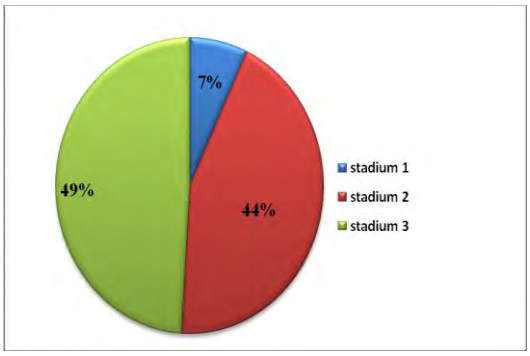
Keterangan : *) jumlah data , **)persentase data, ***) kategori kelas

Tabel 4.4 menunjukan bahwa pada data mayor thyroid (kelas 1) dieliminasi sebanyak 4 data yang dideteksi merupakan kasus tomek links. Pada data mayor kanker payudara (kelas 3) dieliminasi sebanyak 24 data yang dideteksi merupakan kasus tomek links. Pada data mayor kanker serviks (kelas 2) dieliminasi sebanyak 176 data yang dideteksi merupakan kasus tomek links.

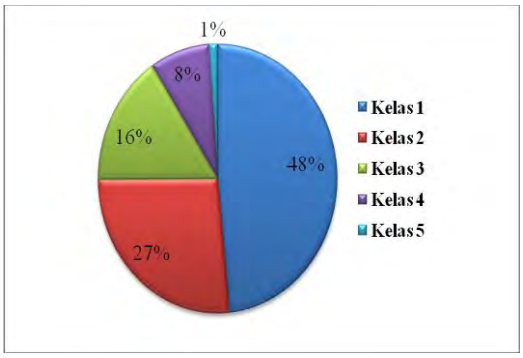
Ilustrasi distribusi data setelah dilakukan Tomek Links dapat dilihat pada Gambar 4.7.



(a) Distribusi Kelas Thyroid



(b) Distribusi Kelas Kanker Payudara



(c) Distribusi Kelas Kanker Serviks

Gambar 4.7 Distribusi Kelas Setelah Tomek Links (a) Data Thyroid; (b) Data Kanker Payudara (c) Data Kanker Serviks

Pada data thyroid setelah dilakukan Tomek Links yaitu pasien dengan kondisi thyroid normal sebesar 72%, kondisi pasien hypothyroid sebesar 16% dan kondisi pasien hyperthyroid sebesar 12%, diilustrasikan pada Gambar 4.7(a). Pada data kanker payudara setelah dilakukan Tomek Links yaitu pasien dengan stadium I sebesar 7%, pasien dengan stadium II sebesar 44%, pasien dengan stadium III sebesar 49%, diilustrasikan pada Gambar 4.7(b). Pada data kanker serviks setelah dilakukan SMOTE yaitu pasien dengan kondisi kelas I sebesar 27%, pasien

dengan kondisi kelas 2 sebesar 48%, kondisi kelas 3 sebesar 16%, kondisi kelas 4 sebesar 8% dan kondisi kelas 5 sebesar 1%.

4.2.3.3 Metode Combine Sampling

Metode combine sampling merupakan perpaduan metode *oversampling* dan *undersampling* yaitu antara metode SMOTE dan Tomek Links. Penggunaan kedua metode ini dilakukan secara berurutan yaitu penanganan menggunakan SMOTE terlebih dahulu selanjutnya hasil SMOTE dilanjutkan menggunakan penanganan Tomek Links. Hasil dari penanganan dengan metode combine sampling ini terhadap masing-masing data *imbalanced* yang digunakan dalam penelitian ini diilustrasikan seperti pada Gambar 4.8. Deskripsi data sebelum dan setelah Combine Sampling dapat dilihat pada Tabel 4.5.

Tabel 4.5 Deskripsi Distribusi Data Sebelum dan Setelah Combine Sampling

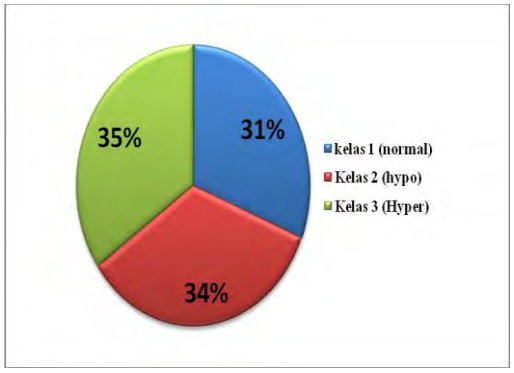
Data	Kelas Mayor hasil SMOTE	Kelas Minor hasil SMOTE	Data Mayor Hapus	Kelas Mayor Baru	Kelas Minor Baru
Thyroid	(168*)(35%**) (3***)	(165*)(34%**) (2***) (150*)(31%**) (1***)	0	(168*)(35%**) (3***)	(165*) (34%**) (2***) (150**)(31%***) (1***)
Kanker Payudara	(134*)(39%**) (2***)	(110*)(32%**) (1***) 100 (29%) (3)	52	(82*)(28%**) (2***)	(110*) (38%**) (1***) 100(34%) (3)
Kanker Serviks	(392*)(23%**) (3***)	(343*)(20%**) (5***) (350*)(20%**) (4***) (340*)(20%**) (2***) (299*)(17%**) (1***)	44 -	(348*)(21%**) (3***)	(343*)(20%**) (5***) (350*)(21%**) (4***) (340*)(20%**) (2***) (299*)(18%**) (1***)

Keterangan : *) jumlah data , **)persentase data, ***) kategori kelas

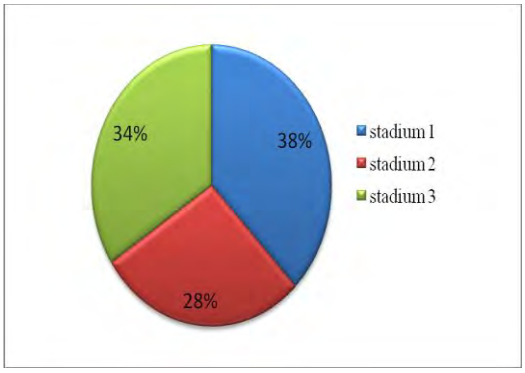
Tabel 4.5 menunjukan bahwa pada data mayor thyroid (kelas 3) hasil SMOTE tidak ada yang dieliminasi atau tidak ada yang terdeteksi kasus tomek links. Pada data mayor kanker payudara (kelas 2) hasil SMOTE dieliminasi sebanyak 52 data yang dideteksi merupakan kasus tomek links. Pada data mayor

kanker serviks (kelas 3) dieliminasi sebanyak 44 data yang dideteksi merupakan kasus tomek links.

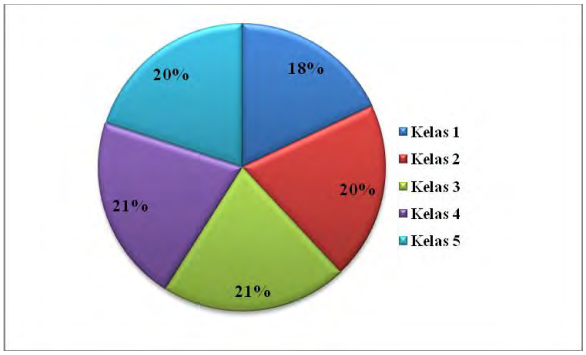
Ilustrasi distribusi data setelah dilakukan Combine Sampling dapat dilihat pada Gambar 4.8.



(a) Distribusi Kelas Thyroid



(b) Distribusi Kelas Kanker Payudara



(c) Distribusi Kelas Kanker Serviks

Gambar 4.8 Distribusi Kelas Setelah Combine Sampling (a) Data Thyroid; (b) Data Kanker Payudara (c) Data Kanker Serviks

Pada data thyroid setelah dilakukan SMOTE yaitu pasien dengan kondisi thyroid normal sebesar 31%, kondisi pasien hypothyroid sebesar 34% dan kondisi pasien hyperthyroid sebesar 35%, diilustrasikan pada Gambar 4.8(a). Pada data kanker payudara setelah dilakukan SMOTE yaitu pasien dengan stadium I sebesar 34%, pasien dengan stadium II sebesar 28%, pasien dengan stadium III sebesar 38%, diilustrasikan pada Gambar 4.8(b). Pada data kanker serviks setelah dilakukan SMOTE yaitu pasien dengan kondisi kelas I sebesar 18%, pasien dengan kondisi kelas 2 sebesar 20%, kondisi kelas 3 sebesar 21%, kondisi kelas 4 sebesar 21% dan kondisi kelas 5 sebesar 20%, diilustrasikan pada Gambar 4.8(c).

4.2.4 Klasifikasi *Multiclass* LS-SVM One Againt One (One)

Data yang sudah seimbang diklasifikasikan menggunakan LS-SVM dan LS-SVM PSO-GSA. Klasifikasi pada penelitian ini merupakan kasus klasifikasi *multiclass*, sehingga menggunakan pendekatan *One Againt One (OAO)*. Algoritma OAO ditambahkan pada algoritma LS-SVM. Pada pendekatan OAO diperoleh $v(v-1)/2$ fungsi pemisah, dimana v adalah banyaknya kelas. Ilustrasi menggunakan pendekatan OAO pada data thyroid yaitu data thyroid memiliki 3 kelas berarti memiliki 3 fungsi pemisah juga. Fungsi pemisah tersbut adalah fungsi pemisah kelas 1 dan kelas 2, fungsi pemisah kelas 2 dan kelas 3, fungsi pemisah kelas 1 dan kelas 3. Ketika melakukan training pada fungsi pemisah kelas 1 dan kelas 2 yaitu jika melakukan *training* pada kelas 1 maka semua anggota pada kelas 1 diberi label 1 sedangkan anggota pada kelas 2 diberi label (-1). Begitu pula ketika melakukan training pada fungsi pemisah antara kelas 1 dan kelas 3 serta kelas 2 dan kelas 3. Untuk dapat kanker payudara sama seperti pada data thyroid yaitu memiliki 3 fungsi pemisah. Untuk data kanker serviks yang memiliki 5 kelas maka memiliki fungsi pemisah sebanyak 10. Prosedur pemberian label sama seperti yang dilakukan pada data thyroid.

Pada sub bab 4.4 akan dipaparkan hasil klasifikasi thyroid, kanker payudara dan kanker serviks dengan LS-SVM dan LS-SVM PSO-GSA pada kondisi data sebelum dan sesudah menggunakan penanganan *imbalanced* data (SMOTE, Tomek Links dan Combine Sampling). Untuk pembagian data training dan testing digunakan q-fold *cross validation* dengan q=5 dan q=10.

4.4.1 Klasifikasi *Multiclass* LS-SVM dan LS-SVM PSO-GSA Untuk 5 Fold

Pada klasifikasi LS-SVM, parameter kernel RBF (σ) dan (C) dilakukan *trial error*. *Trial error* dilakukan sebanyak 9 kali percobaan. Parameter kernel RBF (σ) yang dicobakan yaitu $(\sigma = 1,10,20)$ dan parameter (C) yang dicobakan yaitu $(C = 1,50,100)$.

Pada klasifikasi LS-SVM PSO-GSA, parameter kernel RBF (σ) dan (C) dioptimasi tidak menggunakan *trial error* dan berada pada range yang ditentukan. Penentuan range akan menentukan besar kecilnya akurasi. Parameter kernel RBF

(σ) yang dicobakan yaitu (range $\sigma = 1-20$) dan parameter (C) yang dicobakan yaitu (range $C = 1-100$).

Hasil akurasi (performansi) klasifikasi LS-SVM dan LS-SVM PSO-GSA dari *training* dan *testing* dapat dilihat pada Tabel 4.6-4.17.

Tabel 4.6 Akurasi Klasifikasi Data Training dengan LS-SVM Original 5 Fold

Data	Parameter		Fold					
	C	σ	1	2	3	4	5	Rata-rata
Thyroid		1	98,795	98,193	98,193	98,193	99,39	98,553
		10	97,181	95,181	94,578	93,976	98,171	95,817
	1	20	91,566	92,169	92,771	92,771	97,561	93,368
		1	**100****	**100****	**100****	**100****	**100****	**100,000***
		10	97,59	98,193	97,59	97,59	100	98,193
	50	20	96,988	98,193	96,385	97,59	99,39	97,709
		1	**100****	**100****	**100****	**100****	**100****	**100,000***
		10	99,193	98,193	97,59	98,795	100	98,754
	100	20	96,988	98,193	97,59	97,59	99,39	97,950
Kanker Payudara	1	1	94,366	95,07	94,366	95,07	93,75	94,524
		10	88,732	88,732	88,732	87,324	90,278	88,760
		20	87,324	88,732	88,732	86,619	90,278	88,337
	50	1	95,07	**95,775****	**95,775****	95,07	95,139	**95,366***
		10	94,366	95,07	95,07	94,366	94,444	94,663
		20	92,667	94,366	94,366	92,958	93,75	93,621
	100	1	95,07	**95,775****	**95,775****	95,07	95,139	**95,366***
		10	94,366	95,07	95,07	94,366	94,444	94,663
		20	92,958	94,366	94,366	93,662	93,75	93,820
Kanker Serviks	1	1	73,858	70,866	74,331	74,016	73,899	73,394
		10	52,126	54,016	54,331	51,811	53,459	53,149
		20	49,291	50,079	50,236	49,449	49,843	49,780
	50	1	91,024	88,504	88,346	87,402	88,365	88,728
		10	63,622	65,039	65,512	64,724	63,05	64,389
		20	59,055	57,48	58,583	58,583	58,805	58,501
	100	1	**92,126****	89,449	89,921	88,819	89,779	**90,019***
		10	66,457	66,772	68,031	67,087	66,352	66,940
		20	60,315	59,213	60,472	59,842	60,535	60,075

* : Rata-rata tertinggi

** : Akurasi tertinggi

Tabel 4.6 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 50,100 dengan persentase nilai ketepatan klasifikasinya yaitu 100%. Akurasi klasifikasi tertinggi pada data thyroid dihasilkan pada fold 1,2,3,4,5 dengan nilai σ sebesar 1 dan nilai C sebesar 50,100 yaitu 100%. Pada data kanker payudara

rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 50,100 dengan persentase nilai ketepatan klasifikasinya yaitu 95,366%.

Untuk validasi model klasifikasi digunakan data testing dengan (σ) dan (C) yang telah diperoleh selama proses training. Akurasi klasifikasi pada data testing ditunjukan pada Tabel 4.7

Tabel 4.7 Akurasi Klasifikasi Data Testing dengan LS-SVM Original 5 Fold

Data	Parameter		Fold (%)					Rata-rata
	C	σ	1	2	3	4	5	
Thyroid	1	1	**100****	95,122	**100****	87,805	6,977	77,981
		10	**100****	**100****	**100****	87,805	30,233	83,608
		20	**100****	**100****	**100****	87,805	23,256	82,212
	50	1	97,561	95,122	**100****	90,244	25,581	81,702
		10	**100****	95,122	**100****	95,122	37,209	85,491
		20	**100****	**100****	**100****	95,122	37,209	**86,466***
	100	1	97,561	95,122	**100****	90,244	25,581	81,702
		10	**100****	95,122	**100****	95,122	32,558	84,560
		20	**100****	95,122	**100****	92,683	37,209	85,003
Kanker Payudara	1	1	88,889	88,889	77,778	83,333	82,353	84,248
		10	91,667	86,111	86,111	94,444	79,412	87,549
		20	91,667	86,111	77,778	**97,222****	79,412	86,438
	50	1	88,889	72,222	72,222	83,333	85,294	80,392
		10	91,667	77,778	86,111	86,111	82,353	84,804
		20	91,667	88,889	91,667	91,667	82,353	**89,249***
	100	1	88,889	72, 222	72,222	83,333	85,294	82,435
		10	91,667	86,111	86,111	86,111	82,353	86,471
		20	91,667	88,889	91,667	88,889	82,353	88,693
Kanker Serviks	1	1	43,396	47,799	49,685	43,396	46,835	46,222
		10	44,025	44,541	42,138	**50,314****	46,202	45,444
		20	42,767	44,025	43,396	47,799	44,937	44,585
	50	1	38,365	43,396	40,880	43,396	36,076	40,423
		10	43,396	46,541	45,912	44,025	46,202	45,215
		20	44,025	47,799	46,541	47,799	47,469	**46,727***
	100	1	38,365	41,509	40,88	41,509	33,544	39,161
		10	40,88	45,912	45,283	43,396	48,101	44,714
		20	43,396	46,541	45,283	44,025	47,468	45,343

* : Rata-rata tertinggi

** : Akurasi tertinggi

Tabel 4.7 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 20 dan nilai C sebesar 50 dengan persentase nilai ketepatan klasifikasinya yaitu 86,466%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan

nilai σ sebesar 20 dan nilai C sebesar 50 dengan persentase nilai ketepatan klasifikasinya yaitu 89,249%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 20 dan nilai C sebesar 50 dengan persentase nilai ketepatan klasifikasinya yaitu 46,727%.

Tabel 4.8 Akurasi Klasifikasi Data Traning dengan SMOTE LS-SVM 5 Fold

Data	Parameter		Fold					Rata-rata
	C	σ	1	2	3	4	5	
Thyroid	1	1	99,741	98,964	99,223	99,482	99,227	99,327
		10	98,964	98,964	98,446	96,632	96,392	97,880
		20	98,445	99,223	97,15	96,114	96,134	97,413
	50	1	**100****	99,482	**100****	**100****	99,742	99,845
		10	99,223	99,223	99,482	99,741	99,484	99,431
		20	98,964	98,964	99,482	99,482	98,7111	99,121
	100	1	**100****	**100****	**100****	**100****	99,742	**99,948***
		10	99,741	99,482	99,482	99,741	99,484	99,586
		20	98,964	99,223	99,482	99,482	98,969	99,224
Kanker Payudara	1	1	95,273	94,182	94,545	95,273	93,841	94,623
		10	90,909	88	90,909	93,454	90,942	90,843
		20	89,091	87,273	88,723	91,636	90,217	89,388
	50	1	**96,000****	94,909	94,545	95,636	95,289	**95,276***
		10	93,454	92,364	93,091	94,909	93,841	93,532
		20	92,364	92	93,091	94,909	91,304	92,734
	100	1	**96,000****	94,909	94,545	95,636	95,289	**95,276***
		10	94,545	93,09	94,545	94,909	94,203	94,258
		20	92,364	92,727	93,091	94,909	92,09	93,036
Kanker Serviks	1	1	83,974	84,119	81,073	79,84	75,869	80,975
		10	60,914	60,043	58,883	58,303	48,188	57,266
		20	57,505	57,215	54,749	55,112	45,072	53,931
	50	1	96,157	95,141	92,453	91,08	90,217	93,010
		10	72,734	73,096	71,574	70,341	62,246	69,998
		20	67,73	68,383	66,57	64,032	56,522	64,647
	100	1	**96,592****	95,939	93,401	92,023	91,449	**93,881***
		10	75,272	74,764	72,661	72,444	65,435	72,115
		20	69,471	69,833	67,73	66,497	58,551	66,416

* : Rata-rata tertinggi

** : Akurasi tertinggi

Tabel 4.8 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi training tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C

sebesar 100 yaitu 99,948%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 50,100 yaitu **95,276**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **93,881**%.

Tabel 4.9 Akurasi Klasifikasi Data Testing dengan SMOTE LS-SVM 5 Fold

	Parameter		Fold					Rata-rata
	C	σ	1	2	3	4	5	
Thyroid	1	1	89,691	96,907	**100****	**100****	96,842	96,688
		10	85,567	98,969	98,969	93,814	84,21	92,306
		20	81,443	98,969	96,907	91,753	81,053	90,025
	50	1	94,845	97,938	**100****	**100****	**100****	**98,557***
		10	84,536	96,907	**100****	**100****	96,842	95,657
		20	84,536	96,907	**100****	**100****	92,632	94,815
	100	1	94,845	97,938	**100****	**100****	**100****	**98,557***
		10	85,567	96,907	**100****	**100****	96,842	95,863
		20	84,536	96,907	**100****	**100****	94,737	95,236
Kanker Payudara	1	1	91,304	88,406	**97,101****	86,956	94,118	**91,577***
		10	86,956	81,159	94,203	86,956	86,765	87,208
		20	82,608	78,261	94,203	68,116	85,294	81,696
	50	1	85,507	86,956	95,652	88,406	94,118	90,128
		10	91,304	84,058	92,754	86,956	94,118	89,838
		20	86,956	85,507	94,203	86,956	85,294	87,783
	100	1	85,507	86,956	95,652	88,406	94,118	90,128
		10	92,754	86,956	92,754	86,956	95,588	91,002
		20	86,956	84,058	94,203	86,956	89,706	88,376
Kanker Serviks	1	1	39,42	42,319	47,826	19,71	2,236	30,302
		10	24,348	21,739	14,783	8,696	2,326	14,378
		20	19,13	19,42	13,913	11,014	2,326	13,161
	50	1	43,768	50,145	66,377	48,406	83,721	58,483
		10	31,014	31,594	28,116	13,333	6,686	22,149
		20	29,855	27,536	23,768	9,855	1,453	18,493
	100	1	45,217	49,565	65,217	50,145	**86,917****	**59,412***
		10	34,783	33,043	32,464	15,072	13,081	25,689
		20	28,406	28,116	23,188	11,884	1,453	18,609

* : Rata-rata tertinggi

** : Akurasi tertinggi

Tabel 4.9 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi testing tertinggi yaitu **98,557**%. Pada data kanker payudara rata-rata

persentase ketepatan klasifikasi tertinggi yaitu **91,577**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi yaitu **59,412**%.

Tabel 4.10 Akurasi Klasifikasi Data Traning dengan Tomek Links LS-SVM 5 Fold

	Parameter		Fold					Rata-rata
	C	σ	1	2	3	4	5	
Thyroid	1	1	99,386	98,159	98,773	98,159	99,390	98,773
		10	95,092	95,092	94,478	93,865	98,171	95,340
		20	91,411	92,024	92,638	92,638	98,171	93,376
	50	1	**100****	**100****	**100****	**100****	**100****	**100,000***
		10	97,546	98,159	97,546	98,159	**100****	98,282
		20	97,546	98,159	96,319	96,932	99,390	97,669
	100	1	**100****	**100****	**100****	**100****	**100****	**100,000***
		10	98,773	98,773	98,159	98,773	**100****	98,896
		20	97,546	98,159	97,546	97,546	99,390	98,037
Kanker Payudara	1	1	98,374	97,561	97,561	99,187	96,774	97,891
		10	91,869	90,244	91,869	91,057	93,548	91,717
		20	90,244	90,244	91,057	90,244	93,548	91,067
	50	1	99,187	99,187	**100****	**100****	99,193	99,513
		10	98,374	96,748	98,374	99,187	97,581	98,053
		20	97,561	96,748	96,748	99,187	96,774	97,404
	100	1	99,187	99,187	**100****	**100****	99,199	**99,515***
		10	98,374	97,561	99,187	99,187	97,581	98,378
		20	97,561	96,748	97,561	99,187	97,581	97,728
Kanker Serviks	1	1	78,947	74,696	77,935	78,138	78,831	77,709
		10	59,514	59,103	60,526	60,526	61,694	60,273
		20	57,895	56,883	58,299	58,299	59,073	58,090
	50	1	93,927	92,915	92,713	91,700	92,742	92,799
		10	69,556	69,635	72,065	71,457	69,556	70,454
		20	64,170	64,575	67,611	65,182	64,919	65,291
	100	1	**95,344***	94,534	93,725	93,725	93,347	**94,135***
		10	72,469	72,862	73,482	73,889	73,185	73,177
		20	65,992	65,587	68,826	51,613	66,532	63,710

* : Rata-rata tertinggi

** : Akurasi tertinggi

Tabel 4.10 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi training tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 50,100 yaitu 100%. Pada data kanker payudara rata-rata persentase

ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **99,515**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **94,135**%.

Tabel 4.11 Akurasi Klasifikasi Data Testing dengan Tomek Links LS-SVM 5 Fold

	Parameter		Fold					Rata-rata
	C	σ	1	2	3	4	5	
Thyroid	1	1	100,000	97,561	**100****	78,049	10,000	77,122
		10	100,000	100,000	**100****	90,244	27,500	83,549
		20	100,000	100,000	**100****	82,927	22,500	81,085
	50	1	97,561	97,561	**100****	85,366	20,000	80,098
		10	100,000	97,561	**100****	95,122	35,000	85,537
		20	100,000	100,000	**100****	92,683	32,500	85,037
	100	1	97,561	97,561	**100****	85,366	20,000	80,098
		10	100,000	97,561	**100****	95,122	35,000	85,537
		20	100,000	97,561	**100****	97,561	35,000	**86,024***
Kanker Payudara	1	1	93,548	93,548	90,326	90,323	83,333	90,216
		10	93,548	90,323	87,097	93,548	80,000	88,903
		20	93,548	93,548	80,645	96,774	80,000	88,903
	50	1	93,548	83,871	87,097	87,097	83,333	86,989
		10	96,774	90,322	96,774	90,323	86,667	92,172
		20	96,774	96,774	100,000	90,322	86,667	94,107
	100	1	93,548	83,871	87,097	87,097	83,333	86,989
		10	96,774	90,323	96,774	90,322	86,667	92,172
		20	96,774	90,323	100,000	90,323	86,667	**92,817***
Kanker Serviks	1	1	53,226	53,226	51,613	49,193	53,279	52,107
		10	**58,871****	56,452	53,226	58,064	53,279	55,978
		20	58,064	56,452	54,032	**58,871****	53,279	**56,140***
	50	1	41,129	49,194	44,355	48,387	43,4426	45,302
		10	53,279	50,000	50,000	52,419	53,279	51,795
		20	57,258	55,645	52,419	56,452	50,819	54,519
	100	1	41,129	48,387	44,355	46,774	43,443	44,818
		10	53,279	51,613	49,193	51,613	51,639	51,467
		20	54,839	52,419	52,419	54,032	53,279	53,398

* : Rata-rata tertinggi

** : Akurasi tertinggi

Tabel 4.11 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi testing tertinggi yaitu **86,024**%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi tertinggi yaitu **92,817**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi yaitu **56,140**%.

Tabel 4.12 Akurasi Klasifikasi Data Traning dengan Combine Sampling LS-SVM 5 Fold

	Parameter		Fold					Rata-rata
	C	σ	1	2	3	4	5	
Thyroid	1	1	99,741	98,964	99,223	99,482	99,227	99,327
		10	98,964	98,964	98,446	96,632	96,392	97,880
		20	98,445	99,223	97,150	96,114	96,134	97,413
	50	1	**100****	99,482	**100****	**100****	99,742	99,845
		10	99,223	99,223	99,482	99,741	99,484	99,431
		20	98,964	98,964	99,482	99,482	98,711	99,121
	100	1	**100****	**100****	**100****	**100****	99,742	**99,948***
		10	99,741	99,482	99,482	99,741	99,484	99,586
		20	98,964	99,223	99,482	99,482	98,969	99,224
Kanker Payudara	1	1	99,573	97,863	97,863	98,718	96,983	98,200
		10	91,453	89,743	90,171	94,444	91,379	91,438
		20	90,598	88,461	88,889	94,444	86,207	89,720
	50	1	**100****	99,573	96,573	**100****	99,569	99,143
		10	97,436	96,154	96,154	98,291	94,397	96,486
		20	94,444	94,444	94,872	98,291	92,241	94,858
	100	1	**100****	99,573	99,573	**100****	99,569	**99,743***
		10	99,145	97,436	97,863	98,291	96,121	97,771
		20	95,299	94,444	94,872	98,291	92,672	95,116
Kanker Serviks	1	1	84,896	84,449	81,324	80,283	76,637	81,518
		10	59,896	59,300	59,821	59,152	48,512	57,336
		20	56,548	57,292	55,357	55,357	45,610	54,033
	50	1	96,800	95,908	93,601	91,964	91,220	93,899
		10	73,958	73,958	72,545	71,429	63,467	71,071
		20	68,601	68,899	68,155	64,583	55,804	65,208
	100	1	**97,173****	96,726	94,420	93,006	92,634	**94,792***
		10	75,446	76,265	73,884	73,586	66,815	73,199
		20	70,015	70,015	69,196	67,187	58,333	66,949

* : Rata-rata tertinggi

** : Akurasi tertinggi

Tabel 4.12 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi training tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **99,948**%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **99,743**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **94,792**%.

Tabel 4.13 Akurasi Klasifikasi Data Testing dengan Combine Sampling LS-SVM 5 Fold

	Parameter		Fold					Rata-rata
	C	σ	1	2	3	4	5	
Thyroid	1	1	89,691	96,907	**100****	**100****	96,842	96,688
		10	85,567	98,969	98,969	93,814	84,210	92,306
		20	81,443	98,969	96,907	91,753	81,053	90,025
	50	1	94,845	97,938	**100****	**100****	**100****	**98,557***
		10	84,536	96,907	**100****	**100****	96,842	95,657
		20	84,536	96,907	**100****	**100****	92,632	94,815
	100	1	94,845	97,938	**100****	**100****	**100,000**	**98,557***
		10	85,567	96,907	**100****	**100****	96,842	95,863
		20	84,536	96,907	**100****	**100****	94,737	95,236
Kanker Payudara	1	1	91,379	93,103	98,276	84,483	96,667	**92,782***
		10	84,483	86,207	94,828	84,483	86,667	87,334
		20	82,759	84,483	94,828	84,483	68,333	82,977
	50	1	91,379	87,931	96,552	86,207	**100****	92,414
		10	89,655	87,931	94,828	84,483	95,000	90,379
		20	86,207	86,207	98,276	84,483	90,000	89,035
	100	1	91,379	87,931	96,552	86,207	**100****	92,414
		10	93,103	89,655	94,828	84,483	98,333	92,080
		20	87,931	86,207	98,276	84,483	91,667	89,713
Kanker Serviks	1	1	39,583	42,262	41,667	14,583	0,000	27,619
		10	22,321	21,131	14,286	5,655	0,000	12,679
		20	18,454	18,155	12,202	5,655	0,000	10,893
	50	1	45,238	51,488	65,476	39,286	83,929	57,083
		10	30,655	32,738	26,786	10,119	5,059	21,071
		20	28,571	27,976	23,512	6,250	0,000	17,262
	100	1	45,536	50,595	63,095	42,262	**87,202****	**57,738***
		10	32,738	33,333	30,655	11,012	11,607	23,869
		20	28,869	29,464	23,809	8,631	0,000	18,155

* : Rata-rata tertinggi

** : Akurasi tertinggi

Tabel 4.13 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi testing tertinggi yaitu **98,557**%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi tertinggi yaitu **92,782**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi yaitu **57,738**%.

Pembahasan sebelumnya telah memaparkan klasifikasi multiclass dengan menggunakan LS-SVM. Selanjutnya akan dipaparkan klasifikasi multiclass dengan menggunakan LS-SVM PSO-GSA. Pada klasifikasi LS-SVM PSO-GSA, parameter kernel RBF (σ) dan (C) dioptimasi tidak menggunakan *trial error* dan berada pada range yang ditentukan. Penentuan range akan menentukan besar kecilnya akurasi. Parameter kernel RBF (σ) yang dicobakan yaitu (range $\sigma = 1-20$) dan parameter (C) yang dicobakan yaitu (range $C = 1-100$). Hasil klasifikasi dengan menggunakan LS-SVM PSO-GSA ditunjukan pada Tabel 4.14.

Tabel 4.14 Akurasi Klasifikasi dengan LS-SVM PSO-GSA 5 Fold

Metode	Data	C	σ	Rata-rata Akurasi (%)	
				Training	Testing
LS-SVM PSO-GSA Original	Thyroid	48,58	1,27	100	81,702
	Kanker Payudara	11,00	1,61	94,944	88,137
	Kanker Serviks	100	1	93,104	38,656
LS-SVM PSO-GSA SMOTE	Thyroid	76,40	1,01	99,793	**98,558***
	Kanker Payudara	71,29	5,31	96,914	90,638
	Kanker Serviks	100	1	93,881	59,471
LS-SVM PSO-GSA Tomek	Thyroid	57,56	1,93	100	86,024
	Kanker Payudara	33,68	2,82	99,351	93,102
	Kanker Serviks	99,89	1	94,742	56,378
LS-SVM PSO-GSA Combine	Thyroid	69,50	1	99,793	**98,558***
	Kanker Payudara	48,98	1	99,657	**95,623***
	Kanker Serviks	100	1	95,074	**59,621***

Ket : *) Rata-rata Akurasi 5Fold

Tabel 4.14 menunjukkan bahwa akurasi total 5 fold yang tertinggi pada data ketiga data yaitu menggunakan metode LS-SVM PSO-GSA Combine. Akurasi total untuk mendiagnosis penyakit tiroid sebesar 100%, akurasi total untuk mendiagnosis penyakit kanker payudara sebesar 100% dan akurasi total untuk mendiagnosis penyakit kanker serviks sebesar 59,621%.

Setalah dilakukan pengklasifikasian dengan metode LS-SVM Sebelum dilakukan *imbalanced* dan setelah dilakukan imbalanced baik menggunakan optimasi PSO-GSA maupun dengan trial error, nilai rata-rata tertinggi pada setiap metode terangkum pada Tabel 4.15 sampai dengan 4.21.

Tabel 4.15 Rangkuman nilai rata-rata Akurasi tertinggi Pada Training (q=5)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	**100*** (C=50,100) (σ=1)	99.948 (C=100) (σ=1)	**100*** (C=50,100) (σ=1)	99,948 (C=100) (σ=1)	**100*** (C=48,58) (σ=1,27)	99,793 (C=76,40) (σ=1,01)	**100*** (C=57,56) (σ=1,93)	99,793 (C=69,50) (σ=1)
2	95,366 (C=50,100) (σ=1)	95,276 (C=50,100) (σ=1)	99,515 (C=100) (σ=1)	**99,743*** (C=100) (σ=1)	94,944 (C=11 (σ=1,61)	96,914 (C=71,29) (σ=5,31)	99,351 (C=33,68) (σ=2,82)	99,657 (C=48,98) (σ=1)
3	90,019 (C=100) (σ=1)	93,881 (C=100) (σ=1)	94,135 (C=100) (σ=1)	94,792 (C=100) (σ=1)	93,104 (C=100) (σ=1)	93,881 (C=100) (σ=1)	**99,657*** (C=99,89) (σ=1)	95,074 (C=100) (σ=1)

Ket : *) Rata-rata Akurasi Training Tertinggi

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Tabel 4.15 merupakan rangkuman hasil nilai rata-rata akurasi tertinggi training pada semua metode. Pada data thyroid (1), rata-rata akurasi *training* tertinggi sebesar 100%. Pada data kanker payudara (2), rata-rata akurasi training tertinggi dihasilkan oleh metode *Combine* LS-SVM sebesar 99,743%. Pada data kanker serviks (3), rata-rata akurasi training tertinggi dihasilkan oleh metode Tomek Links LS-SVM PSO-GSA sebesar 99,657%

Untuk validasi model klasifikasi digunakan data testing dengan (σ) dan (C) yang telah diperoleh selama proses *training*. Rangkuman nilai rata-rata Akurasi klasifikasi tertinggi pada data *testing* di semua metode yang dicobakan dapat dilihat pada Tabel 4.16.

Tabel 4.16 Rangkuman nilai rata-rata Akurasi Total Pada Testing (q=5 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	86,466 (C=50) (σ=20)	98,557 (C=50,100) (σ=1)	86,024 (C=100) (σ=20)	98,557 (C=50,100) (σ=1)	81,702 (C=48,58) (σ=1,27)	**98,558*** (C=76,40) (σ=1,01)	86,024 (C=57,56) (σ=1,93)	**98,558*** (C=69,50) (σ=1)
2	89,249 (C=50) (σ=20)	91,577 (C=1) (σ=1)	92,817 (C=100) (σ=20)	92,782 (C=1) (σ=1)	88,137 (C=11) (σ=1,61)	90,638 (C=71,29) (σ=5,31)	93,102 (C=33,68 (σ=2,82)	**95,623*** (C=48,98 (σ=1)
3	46,727 (C=50) (σ=20)	59,412 (C=100) (σ=1)	56,140 (C=1) (σ=20)	57,738 (C=100) (σ=1)	38,656 (C=100) (σ=1)	59,471 (C=100) (σ=1)	56,378 (C=99,89) (σ=1)	**59,621*** (C=100) (σ=1)

Ket : *) Rata-rata Akurasi Tertinggi

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Tabel 4.17 Rangkuman nilai rata-rata *Sensitivity* Pada Testing (q=5 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	86,466 (C=50) (σ=20)	97,083 (C=50,100) (σ=1)	86,024 (C=100) (σ=20)	97,083 (C=50,100) (σ=1)	86,525 (C=48,58) (σ=1,27)	**97,230*** (C=76,40) (σ=1,01)	86,224 (C=57,56) (σ=1,93)	**97,230*** (C=69,50) (σ=1)
2	89,248 (C=50) (σ=20)	91,577 (C=1) (σ=1)	82,817 (C=100) (σ=20)	92,782 (C=1) (σ=1)	82,353 (C=11) (σ=1,61)	92,636 (C=71,29) (σ=5,31)	90,234 (C=33,68) (σ=2,82)	**93,450*** (C=48,98 (σ=1)
3	46,161 (C=50) (σ=20)	59,974 (C=100) (σ=1)	55,513 (C=1) (σ=20)	60,576 (C=100) (σ=1)	37,456 (C=100) (σ=1)	59,898 (C=100) (σ=1)	56,256 (C=99,89) (σ=1)	**61,422*** (C=100) (σ=1)

Ket : *) Rata-rata Sensitivity Tertinggi

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Tabel 4.18 Rangkuman nilai rata-rata *Specificity* Pada Testing (q=5 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	92,185 (C=50) (σ=20)	95,046 (C=50) (σ=20)	91,591 (C=100) (σ=20)	94,974 (C=50,100) (σ=1)	92,165 (C=48,58) (σ=1,27)	**98,120*** (C=76,40) (σ=1,01)	92,450 (C=57,56) (σ=1,93)	**98,120*** (C=69,50) (σ=1)
2	97,974 (C=50,100) (σ=1)	94,781 (C=1) (σ=1)	96,828 (C=100) (σ=20)	94,953 (C=1) (σ=1)	91,781 (C=11 (σ=1,61)	95,657 (C=71,29) (σ=5,31)	96,869 (C=33,68 (σ=2,82)	**98,456*** (C=48,98 (σ=1)
3	94,974 (C=50,100) (σ=1)	64,484 (C=100) (σ=1)	83,244 (C=1) (σ=20)	**84,979*** (C=100) (σ=1)	75,612 (C=100) (σ=1)	65,123 (C=100) (σ=1)	83,423 (C=99,89) (σ=1)	84,213 (C=100) (σ=1)

Ket : *) Rata-rata Specificity Tertinggi

Tabel 4.19 Rangkuman nilai rata-rata P*recision* Pada Testing (q=5 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	86,466 (C=50) (σ=20)	97,083 (C=50,100) (σ=1)	86,024 (C=100) (σ=20)	97,083 (C=50,100) (σ=1)	86,525 (C=48,58) (σ=1,27)	**97,230*** (C=76,40) (σ=1,01)	86,224 (C=57,56) (σ=1,93)	**97,230*** (C=69,50) (σ=1)
2	89,248 (C=50) (σ=20)	91,577 (C=1) (σ=1)	82,817 (C=100) (σ=20)	92,782 (C=1) (σ=1)	82,353 (C=11) (σ=1,61)	92,636 (C=71,29) (σ=5,31)	90,234 (C=33,68) (σ=2,82)	**93,450*** (C=48,98 (σ=1)
3	46,161 (C=50) (σ=20)	59,974 (C=100) (σ=1)	55,513 (C=1) (σ=20)	60,576 (C=100) (σ=1)	37,456 (C=100) (σ=1)	59,898 (C=100) (σ=1)	56,256 (C=99,89) (σ=1)	**61,422*** (C=100) (σ=1)

Ket : *) Rata-rata Precision Tertinggi

Tabel 4.20 Rangkuman nilai rata-rata Fmeasure Pada Testing (q=5 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	86,466 (C=50) (σ=20)	97,083 (C=50,100) (σ=1)	86,024 (C=100) (σ=20)	97,083 (C=50,100) (σ=1)	86,525 (C=48,58) (σ=1,27)	**97,230*** (C=76,40) (σ=1,01)	86,224 (C=57,56) (σ=1,93)	**97,230*** (C=69,50) (σ=1)
2	89,248 (C=50) (σ=20)	91,577 (C=1) (σ=1)	82,817 (C=100) (σ=20)	92,782 (C=1) (σ=1)	82,353 (C=11) (σ=1,61)	92,636 (C=71,29) (σ=5,31)	90,234 (C=33,68) (σ=2,82)	**93,450*** (C=48,98 (σ=1)
3	46,161 (C=50) (σ=20)	59,974 (C=100) (σ=1)	55,513 (C=1) (σ=20)	60,576 (C=100) (σ=1)	37,456 (C=100) (σ=1)	59,898 (C=100) (σ=1)	56,256 (C=99,89) (σ=1)	**61,422*** (C=100) (σ=1)

Ket : *) Rata-rata Fmeasurey Tertinggi

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Tabel 4.21 Rangkuman nilai rata-rata Gmean Pada Testing (q=5 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	88,963 (C=50) (σ=20)	97,525 (C=50,100) (σ=1)	88,430 (C=100) (σ=20)	97,525 (C=50,100) (σ=1)	89,000 (C=48,58) (σ=1,27)	**98,423*** (C=76,40) (σ=1,01)	88,965 (C=57,56) (σ=1,93)	**98,423*** (C=69,50) (σ=1)
2	92,095 (C=50) (σ=20)	93,158 (C=1) (σ=1)	94,791 (C=100) (σ=20)	93,857 (C=1) (σ=1)	91,235 (C=11 (σ=1,61)	93,164 (C=71,29) (σ=5,31)	95,123 (C=33,68 (σ=2,82)	**96,423*** (C=48,98 (σ=1)
3	59,772 (C=50) (σ=20)	70,938 (C=100) (σ=1)	67,968 (C=1) (σ=20)	71,535 (C=100) (σ=1)	50,612 (C=100) (σ=1)	71,102 (C=100) (σ=1)	56,256 (C=99,89) (σ=1)	**72,133*** (C=100) (σ=1)

Ket : *) Rata-rata Gmean Tertinggi

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Tabel 4.16 sampai dengan Tabel 4.21 merupakan rangkuman hasil nilai rata-rata akurasi, *Sensitivity, Specificity, Precision, Fmeasure, Gmean* tertinggi testing pada semua metode. Hasil menunjukkan bahwa metode Combine LS-SVM PSO-GSA merupakan metode terbaik atau unggul di semua data percobaan baik diukur performansi dari akurasi total, *Sensitivity, Specificity, Precision, Fmeasure* dan *Gmean.*

Berdasarkan Hasil dari Tabel 4.16 sampai dengan 4.21, maka telah diketahui metode yang menghasilkan akurasi tertinggi pada setiap data percobaan. Selanjutnya akan dituliskan model persamaan LS-SVM pada setiap data berdasarkan metode terbaiknya. Model disusun berdasarkan nilai rata-rata akurasi tertinggi pada Training.

Pada data thyroid memiliki 3 kelas maka akan terbentuk tiga model persamaan. Persamaan model *Combine* LS-SVM PSO-GSA *multiclass* pada data thyroid dapat ditulis sebagai berikut:

i. Untuk kelas 1 dan kelas 2 (C= 69,50 dan σ=1)

Diketahui :

$x_i=[x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}]$,i=1,2,...,388

$x_j=[x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}]$, j=1,2,...,388

Diperoleh :

$b = 0{,}0433$

α_i berukuran 483×1 ($\alpha_1 = -9{,}4943$; $\alpha_2 = 4{,}0562; ...; \alpha_{388} = 0$)

Maka

$\widehat{f}_1(\mathbf{x}) = (\widehat{\mathbf{w}}^T x+b)$

dimana $\widehat{\mathbf{w}}= \sum_{i=1}^{388}\sum_{j=1}^{388} \alpha_i y_i K\left(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j}\right)$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\|\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right\|^2}{2\sigma^2}\right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 1 dan kelas 2) adalah

$$\widehat{f}_1(\mathbf{x}) = \sum_{i=1}^{388}\sum_{j=1}^{388} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right) + 0{,}0433$$

ii. Untuk kelas 1 dan kelas 3 (C= 69,50 dan σ=1)

Diketahui :

$x_i=[x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}]$,i=1,2,...,388

$x_j=[x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}]$, j=1,2,...,388

Diperoleh :

b = -0,4804

α_i berukuran 388×1 ($\alpha_1 = -1,0536$; $\alpha_2 = -4,0544; ...; \alpha_{388} = -1,5937$)

Maka

$\hat{f}_2(\mathbf{x}) = (\hat{\mathbf{w}}^T x+b)$

dimana $\hat{\mathbf{w}}= \sum_{i=1}^{388}\sum_{j=1}^{388} \alpha_i y_i K(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j})$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\|\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right\|^2}{2\sigma^2}\right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 1 dan kelas 3) adalah

$$\hat{f}_2(\mathbf{x}) = \sum_{i=1}^{388}\sum_{j=1}^{388} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right) - 0,4804$$

iii. Untuk kelas 1 dan kelas 3 (C= 69,50 dan σ=1)

Diketahui :

$x_i=[x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}]$,i=1,2,...,388

$x_j=[x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}]$, j=1,2,...,388

Diperoleh :

b = -0,1508

α_i berukuran 388×1 ($\alpha_1 = 0$; $\alpha_2 = 0; ...; \alpha_{388} = 0,0374$)

Maka

$\hat{f}_3(\mathbf{x}) = (\hat{\mathbf{w}}^T x+b)$

dimana $\hat{\mathbf{w}}= \sum_{i=1}^{388}\sum_{j=1}^{388} \alpha_i y_i K(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j})$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\| \mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right\|^2}{2\sigma^2} \right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 2 dan kelas 3) adalah

$$\widehat{f}_3(\mathbf{x}) = \sum_{i=1}^{388} \sum_{j=1}^{388} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right) - 0{,}1508$$

Pada data Kanker Payudara memiliki 3 kelas maka akan terbentuk tiga model persamaan. Persamaan model *Combine* LS-SVM PSO-GSA *multiclass* pada data kanker payudara dapat ditulis sebagai berikut:

i. Untuk kelas 1 dan kelas 2 (C= 48,98 dan σ=1)

Diketahui :

$x_i = [x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}, x_{6i}]$,i=1,2,...,292

$x_j = [x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}, x_{6i}]$, j=1,2,...,292

Diperoleh :

b = 0,3417

α_i berukuran 292×1 ($\alpha_1 = -0{,}6195$; $\alpha_2 = 0; ...; \alpha_{292} = -1{,}36744$)

Maka

$\widehat{f}_1(\mathbf{x}) = (\widehat{\mathbf{w}}^T \text{x+b})$

dimana $\widehat{\mathbf{w}} = \sum_{i=1}^{292} \sum_{j=1}^{292} \alpha_i y_i K\left(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j} \right)$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\| \mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right\|^2}{2\sigma^2} \right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 1 dan kelas 2) adalah

$$\widehat{f}_1(\mathbf{x}) = \sum_{i=1}^{292} \sum_{j=1}^{292} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right) + 0{,}3417$$

ii. Untuk kelas 1 dan kelas 3 (C= 48,98 dan σ=1)

Diketahui :

$x_i=[x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}, x_{6i}]$,i=1,2,...,292

$x_j=[x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}, x_{6i}]$, j=1,2,...,292

Diperoleh :

b = -0,5887

α_i berukuran 292×1 ($\alpha_1 = 0$; $\alpha_2 = -4,593; ...; \alpha_{292} = 0$)

Maka

$\hat{f}_1(\mathbf{x}) = (\hat{\mathbf{w}}^T x+b)$

dimana $\hat{\mathbf{w}}= \sum_{i=1}^{292}\sum_{j=1}^{292} \alpha_i y_i K\left(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j}\right)$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\|\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right\|^2}{2\sigma^2}\right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 1 dan kelas 3) adalah

$$\hat{f}_1(\mathbf{x}) = \sum_{i=1}^{292}\sum_{j=1}^{292} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right) - 0,5887$$

iii. Untuk kelas 2 dan kelas 3 (C= 48,98 dan σ=1)

Diketahui :

$x_i=[x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}, x_{6i}]$,i=1,2,...,292

$x_j=[x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}, x_{6i}]$, j=1,2,...,292

Diperoleh :

b = -0,3883

α_i berukuran 292×1 ($\alpha_1 = 0,4105$; $\alpha_2 = 4,3462; ...; \alpha_{292} = 1,43871$)

Maka

$\hat{f}_1(\mathbf{x}) = (\hat{\mathbf{w}}^T x+b)$

dimana $\hat{\mathbf{w}}= \sum_{i=1}^{292}\sum_{j=1}^{292} \alpha_i y_i K\left(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j}\right)$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\| \mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right\|^2}{2\sigma^2} \right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 2 dan kelas 3) adalah

$$\widehat{f}_1(\mathbf{x}) = \sum_{i=1}^{292} \sum_{j=1}^{292} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right) - 0{,}3883$$

Pada data Kanker serviks memiliki 5 kelas maka akan terbentuk 10 model persamaan. Persamaan model *Combine* LS-SVM PSO-GSA *multiclass* pada data kanker serviks dapat ditulis sebagai berikut:

Untuk kelas 2 dan kelas 3 (C= 100 dan σ=1) adalah

Diketahui :

$\mathrm{x}_i = [\mathrm{x}_{1i}, \mathrm{x}_{2i}, \mathrm{x}_{3i}, \mathrm{x}_{4i}, \mathrm{x}_{5i}, \mathrm{x}_{6i}, \mathrm{x}_{7i}]$,i=1,2,...,558

$\mathrm{x}_j = [\mathrm{x}_{1j}, \mathrm{x}_{2j}, \mathrm{x}_{3j}, \mathrm{x}_{4j}, \mathrm{x}_{5j}, \mathrm{x}_{6i}, \mathrm{x}_{7i}]$, j=1,2,...,558

Diperoleh :

$\mathrm{b} = -0{,}0229$

α_i berukuran 558×1 ($\alpha_1 = -0{,}6195$; $\alpha_2 = 0; \ldots; \alpha_{558} = -1{,}36744$)

Maka

$\widehat{f}_1(\mathbf{x}) = (\widehat{\mathbf{w}}^T \mathrm{x} + \mathrm{b})$

dimana $\widehat{\mathbf{w}} = \sum_{i=1}^{558} \sum_{j=1}^{558} \alpha_i y_i K\left(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j} \right)$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\| \mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right\|^2}{2\sigma^2} \right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 1 dan kelas 2) adalah

$$\widehat{f}_1(\mathbf{x}) = \sum_{i=1}^{558} \sum_{j=1}^{558} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right) - 0{,}0229$$

4.2.5 Perbandingan Kebaikan Metode dengan Uji Friedman

Untuk membuktikan mana metode yang terbaik pada semua data percobaan digunakan pengujian Friedman. Pada uji ini, data dijadikan blok dan perlakuannya adalah metode yang digunakan. Pada masing-masing data dicobakan kedelapan metode tersebut. Ada tiga data yang digunakan dan ada delapan metode yang digunakan. Pengujian Friedman pada q-fold (q=5) adalah sebagai berikut.

(i) Hipotesis

H_0 : $R_1 = R_2 = R_3 = R_4$ (tidak ada metode yang berbeda)

H_1 : minimal ada satu dari R_j berbeda atau tidak sama

(ii) Hipotesis

H_0 : $R_5 = R_6 = R_7 = R_8$ (tidak ada metode yang berbeda)

H_1 : minimal ada satu dari R_j berbeda atau tidak sama

(iii) Hipotesis

H_0 : $R_1 = R_2 = ... = R_8$ (tidak ada metode yang berbeda)

H_1 : minimal ada satu dari R_j berbeda atau tidak sama

Tingkat Signifikan : ($\alpha = 5\%$)

Daerah Kritis : Tolak H_0 jika $\chi^2_{hitung} > \chi^2_{df,\alpha}$ atau p-Value< α

Tabel 4.22 Uji Perbandingan Kebaikan Metode Klasifikasi LS-SVM (q=5) Berdasarkan nilai rata-rata Akurasi Total dengan Uji Friedman

Metode	Chis-Square	Df	p-Value	Keputusan
M1 s/d M4	4,241	3	0,237	Gagal Tolak H_0
M5 s/d M8	7,759	3	0,051	Gagal Tolak H_0
M1 s/d M8	14,028	7	0,051	Gagal Tolak H_0

Ket :

$\chi^2_{3;0,05} = 7,815$; $\chi^2_{7;0,05} = 14,067$

Tabel 4.23 Uji Perbandingan Kebaikan Metode Klasifikasi LS-SVM (q=5) Berdasarkan nilai rata-rata *Sensitivity* dengan Uji Friedman

Metode	Chis-Square	Df	p-Value	Keputusan
M1 s/d M4	7,966	3	0,047*	Tolak H_0
M5 s/d M8	6,517	3	0,089	Gagal Tolak H_0
M1 s/d M8	18,200	7	0,011*	Tolak H_0

Ket : *) Minimal ada salah satu metode yang berbeda pada tingkat signfikan α=5%

$\chi^2_{3;0,05} = 7,815$; $\chi^2_{7;0,05} = 14,067$

Tabel 4.24 Uji Perbandingan Kebaikan Metode Klasifikasi LS-SVM (q=5) Berdasarkan nilai rata-rata *G-Mean* dengan Uji Friedman

Metode	Chis-Square	Df	p-Value	Keputusan
M1 s/d M4	4,655	3	0,199	Gagal Tolak H_0
M5 s/d M8	6,517	3	0,089	Gagal Tolak H_0
M1 s/d M8	12,992	7	0,072	Gagal Tolak H_0

Ket :

$\chi^2_{3;0,05} = 7,815$; $\chi^2_{7;0,05} = 14,067$

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Berdasarkan Tabel 4.22, dengan menggunakan tingkat signfikansi ($\alpha = 5\%$) disimpulkan bahwa pada metode LS-SVM tanpa penanganan *imbalanced* dan menggunakan *imbalanced* data (M1 s/d M4), LS-SVM PSO-GSA tanpa menggunakan penanganan *imbalanced* dan menggunakan *imbalanced* data (M5 s/d M8) serta pada kedelapan metode yang dicobakan (M1 s/d M8) menghasilkan akurasi testing (validasi model) yang sama atau tidak ada metode yang terbaik pada tingkat signifikan 5%.

Berdasarkan Tabel 4.23, dengan menggunakan tingkat signfikansi ($\alpha = 5\%$) disimpulkan bahwa pada metode LS-SVM tanpa penanganan *imbalanced* dan menggunakan *imbalanced* data (M1 s/d M4), LS-SVM PSO-GSA tanpa menggunakan penanganan *imbalanced* dan menggunakan *imbalanced* data (M5 s/d M8) serta pada kedelapan metode yang dicobakan (M1 s/d M8) menghasilkan akurasi *sensitivity* (validasi model) yang berbeda pada tingkat signifikan 5%. Untuk mengetahui dari kedelapan metode tersebut,mana metode yang berbeda maka dilakukan uji pembandingan berganda.

Hipotesis :

H_0 : $R_j = R_{j^*}$ (tidak terdapat perbedaan efek perlakuan j dengan j^*)

H_1 : $R_j \neq R_{j^*}$ (terdapat perbedaan efek perlakuan j dengan j^*) ; j=1,2,…,8

Daerah Kritis : Tolak H_0 jika $\left|R_j - R_{j^*}\right| > Z_{\{1-(\alpha/k(k-1))\}}\sqrt{\frac{bk(k+1)}{6}}$

Dimana :

$$Z_{\{1-(0,05/8(8-1)\}}\sqrt{\frac{3(8)(8+1)}{6}} = 2,36(6) = 14,16$$

Tabel 4.25 Pembandingan Berganda Berdasarkan nilai *Sensitivity*

Jumlah ranking	metode	M1	M2	M3	M4	M5	M6	M7	M8
8	M1	0							
16,5	M2	-8.5							
6	M3	2	10.5						
19,5	M4	-11.5	-3	-13.5					
6	M5	2	10.5	0	13.5				
18,5	M6	-10.5	-2	-12.5	1	-12.5			
10	M7	-2	6.5	-4	9.5	-4	8.5		
23,5	M8	**-15.5***	-7	**-17.5***	-4	**-17.5***	-5	-13.5	

Ket : *) Tolak H_0 pada tingkat signifikan 5%

Berdasarkan Tabel 4.23, dengan menggunakan tingkat signfikansi ($\alpha = 5\%$) disimpulkan bahwa metode yang berbeda adalah metode 1 dan metode 8, metode 3 dan metode 8. Jumlah ranking metode 1 (R_1) lebih kecil dari Jumlah ranking metode 8 (R_8), Jumlah ranking metode 3 (R_3) lebih kecil dari Jumlah ranking metode 8 (R_8) dan Jumlah ranking metode 5 (R_5) lebih kecil dari Jumlah ranking metode 8 (R_8) maka dapat disimpulkan bahwa metode yang terbaik dalam mengukur performansi imbalanced data kelas positif (minor) adalah dengan menggunakan metode 8 (Combine LS-SVM PSO-GSA).

Berdasarkan Tabel 4.24, dengan menggunakan tingkat signfikansi ($\alpha = 5\%$) disimpulkan bahwa pada metode LS-SVM tanpa penanganan *imbalanced* dan menggunakan *imbalanced* data (M1 s/d M4), LS-SVM PSO-GSA tanpa menggunakan penanganan *imbalanced* dan menggunakan *imbalanced* data (M5 s/d M8) serta pada kedelapan metode yang dicobakan (M1 s/d M8) menghasilkan *G-mean testing* (validasi model) yang sama atau tidak ada metode yang terbaik pada tingkat signifikan 5%.

4.2.5 Klasifikasi *Multiclass* LS-SVM dan LS-SVM PSO-GSA Untuk 10 Fold

Seperti halnya pada pembahasan sebelumnya maka pada klasifikasi LS-SVM Q-Fold (q=10) juga akan dilakukan klasifikasi LS-SVM sebelum dilakukan penanganan *imbalanced* data dan setelah dilakukan penanganan *imbalanced* data. Berikut ini merupakan hasil klasifikasi multiclass LS-SVM sebelum dilakukan imbalanced sampai dengan telah dilakukan *imbalanced* data.

Tabel 4.26 Akurasi Klasifikasi Data Training dengan LS-SVM Original

Data	Parameter		Rata-rata	Data	Parameter		Rata-rata	Data	Parameter		Rata-rata
	C	σ			C	σ			C	σ	
Thyroid		1	98,711	Kanker Payudara	1	**1**	94,258	Kanker Serviks	1	1	72,656
		10	95,059			10	88,887			10	53,680
	1	20	92,912			20	88,075			20	49,776
		1	**100,000***		50	1	**95,068***		50	1	87,307
		10	98,281			10	94,444			10	63,252
	50	20	96,832			20	93,632			20	56,055
		1	**100,000***		100	1	72,656		100	1	**88,679***
		10	98,335			10	53,680			10	65,449
	100	20	97,745			20	49,776			20	59,586

* : Rata-rata tertinggi

Tabel 4.27 Akurasi Klasifikasi Data Testing dengan LS-SVM Original

Data	Parameter		Rata-rata	Data	Parameter		Rata-rata	Data	Parameter		Rata-rata
	C	σ			C	σ			C	σ	
Thyroid	1	1	82,381	Kanker Payudara	1	**1**	85,204	Kanker Serviks	1	1	**45,097***
		10	91,429			10	87,431			10	44,447
		20	89,524			20	87,986			20	42,927
	50	1	82,936		50	1	84,236		50	1	40,052
		10	93,118			10	87,569			10	44,682
		20	92,381			20	**89,236***			20	44,988
	100	1	82,636		100	1	83,681		100	1	39,286
		10	**93,333***			10	87,014			10	44,452
		20	92,857			20	88,125			20	43,814

* : Rata-rata tertinggi

Tabel 4.26 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi training tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 50,100 yaitu 100%. Pada data kanker payudara rata-rata persentase

ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 50 yaitu **95,068**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **88,679**%.

Tabel 4.27 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi testing tertinggi saat menggunakan nilai σ sebesar 10 dan nilai C sebesar 100 yaitu 93,333%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi testing tertinggi saat menggunakan nilai σ sebesar 20 dan nilai C sebesar 50 yaitu **89,236**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi testing tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 1 yaitu **45,097**%.

Tabel 4.28 Akurasi Klasifikasi Data Traning dengan SMOTE LS-SVM

Data	Parameter		Rata-rata	Data	Parameter		Rata-rata	Data	Parameter		Rata-rata
	C	σ			C	σ			C	σ	
Thyroid	1	1	99,264	Kanker Payudara	1	**1**	94,509	Kanker Serviks	1	1	80,336
		10	98,320			10	90,438			10	55,587
		20	97,392			20	89,955			20	51,762
	50	1	99,793		50	1	**95,220***		50	1	92,427
		10	99,471			10	93,444			10	68,863
		20	99,333			20	92,884			20	63,237
	100	1	**99,885***		100	1	95,155		100	1	**93,381***
		10	99,448			10	93,928			10	70,861
		20	96,856			20	93,314			20	65,531

* : Rata-rata tertinggi

Tabel 4.28 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi training tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu 99,885%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 50 yaitu **95,220**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **93,381**%.

Untuk validasi model klasifikasi, digunakan data testing yang ditentukan dengan menngunakan q-fold (q=10). Pada data *testing* digunakan nilai σ dan nilai C yang telah diperoleh selama proses training. Akurasi klasifikasi pada data testing ditunjukkan pada Tabel 4.29.

Tabel 4.29 Akurasi Klasifikasi Data Testing dengan SMOTE LS-SVM

Data	Parameter		Rata-rata	Data	Parameter		Rata-rata	Data	Parameter		Rata-rata
	C	σ			C	σ			C	σ	
Thyroid	1	1	97,953	Kanker Payudara	1	**1**	**91,238***	Kanker Serviks	1	1	60,581
		10	95,343			10	88,653			10	35,690
		20	93,934			20	86,857			20	31,665
	50	1	99,167		50	1	90,975		50	1	68,314
		10	97,953			10	89,768			10	53,898
		20	96,740			20	88,947			20	46,434
	100	1	**99,375***		100	1	91,209		100	1	**69,012***
		10	98,346			10	90,356			10	55,174
		20	97,341			20	89,505			20	48,577

* : Rata-rata tertinggi

Tabel 4.29 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi testing tertinggi yaitu **99,375**%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi tertinggi yaitu **91,238**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi yaitu **69,012**%.

Tabel 4.30 Akurasi Klasifikasi Data Traning dengan Tomek Links LS-SVM

Data	Parameter		Rata-rata	Data	Parameter		Rata-rata	Data	Parameter		Rata-rata
	C	σ			C	σ			C	σ	
Thyroid		1	99,076	Kanker Payudara	1	**1**	97,833	Kanker Serviks	1	1	76,717
		10	95,215			10	91,849			10	60,086
	1	20	93,257			20	90,842			20	58,145
		1	**100,000***		50	1	**99,350***		50	1	92,071
		10	98,151			10	97,979			10	69,508
	50	20	97,552			20	97,332			20	63,340
		1	**100,000***		100	1	98,703		100	1	**93,348***
		10	98,640			10	98,051			10	71,916
	100	20	97,878			20	97,476			20	65,965

* : Rata-rata tertinggi

Tabel 4.30 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi training tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 50,100 yaitu 100%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 50 yaitu **99,350**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **93,348**%.

Untuk validasi model klasifikasi, digunakan data testing yang ditentukan dengan menngunakan q-fold (q=10). Pada data testing digunakan nilai σ dan nilai C yang telah diperoleh selama proses training. Akurasi klasifikasi pada data testing ditunjukkan pada Tabel 4.31.

Tabel 4.31 Akurasi Klasifikasi Data Testing dengan Tomek Links LS-SVM

Data	Parameter		Rata-rata	Data	Parameter		Rata-rata	Data	Parameter		Rata-rata
	C	σ			C	σ			C	σ	
Thyroid	1	1	**93,000***	Kanker Payudara	1	**1**	91,087	Kanker Serviks	1	1	52,935
		10	84,500			10	91,228			10	55,677
		20	83,000			20	91,228			20	**56,167**
	50	1	79,500		50	1	92,281		50	1	44,349
		10	87,000			10	**95,474**			10	52,774
		20	87,000			20	**95,474**			20	54,220
	100	1	85,000		100	1	92,281		100	1	44,349
		10	87,500			10	**95,474***			10	50,979
		20	86,000			20	**95,474***			20	53,575

* : Rata-rata tertinggi

Tabel 4.31 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi training tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 1 yaitu 93%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **95,574**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 20 yaitu **56,167**%.

Tabel 4.32 Akurasi Klasifikasi Data Traning dengan Combine Sampling LS-SVM

Data	Parameter		Rata-rata	Data	Parameter		Rata-rata	Data	Parameter		Rata-rata
	C	σ			C	σ			C	σ	
Thyroid		1	99,264	Kanker Payudara	1	**1**	98,020	Kanker Serviks	1	1	80,675
		10	98,465			10	90,526			10	55,622
	1	20	97,392			20	89,916			20	51,693
		1	99,793		50	1	**99,696***		50	1	93,333
		10	99,471			10	95,942			10	69,795
	50	20	99,333			20	95,470			20	64,094
		1	**99,885***		100	1	**99,696***		100	1	**94,386***
		10	99,448			10	97,450			10	72,031
	100	20	96,856			20	95,585			20	66,071

* : Rata-rata tertinggi

Tabel 4.32 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi training tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **99,885**%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 50,100 yaitu **99,696**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **94,386**%.

Untuk validasi model klasifikasi, digunakan data testing yang ditentukan dengan menngunakan q-fold (q=10). Pada data testing digunakan nilai σ dan nilai C yang telah diperoleh selama proses training. Akurasi klasifikasi pada data testing ditunjukkan pada Tabel 4.33.

Tabel 4.33 Akurasi Klasifikasi Data Testing dengan Combine Sampling LS-SVM

Data	Parameter		Rata-rata	Data	Parameter		Rata-rata	Data	Parameter		Rata-rata
	C	σ			C	σ			C	σ	
Thyroid		1	97,953	Kanker Payudara	1	**1**	92,458	Kanker Serviks	1	1	64,762
		10	95,343			10	89,077			10	36,667
	1	20	93,934			20	86,730			20	30,833
		1	99,167		50	1	**93,793***		50	1	68,036
		10	97,953			10	91,769			10	52,679
	50	20	96,740			20	91,491			20	45,060
		1	**99,375***		100	1	93,448		100	1	**68,312***
		10	98,346			10	92,759			10	55,000
	100	20	97,341			20	92,158			20	47,679

* : Rata-rata tertinggi

Tabel 4.33 diketahui bahwa pada data thyroid rata-rata persentase ketepatan klasifikasi training tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu 99,375%. Pada data kanker payudara rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 50 yaitu **93,793**%. Pada data kanker serviks rata-rata persentase ketepatan klasifikasi tertinggi saat menggunakan nilai σ sebesar 1 dan nilai C sebesar 100 yaitu **68,312**%.

Tabel 4.34 Akurasi Klasifikasi dengan LS-SVM PSO-GSA

Metode	Data	C	σ	Akurasi (%)	
				Training	Testing
LS-SVM PSO-GSA Original	Thyroid	80,50	2,99	100	96,774
	Kanker Payudara	44,14	2,36	94,944	89,620
	Kanker Serviks	100	1	90,721	39,919
LS-SVM PSO-GSA SMOTE	Thyroid	76,75	1	99,793	**99,378***
	Kanker Payudara	63,60	2,85	95,058	92,432
	Kanker Serviks	100	1	93,902	59,554
LS-SVM PSO-GSA Tomek	Thyroid	61,05	1	100	95,109
	Kanker Payudara	73,03	2,84	99,351	95,666
	Kanker Serviks	100	1	93,348	44,672
LS-SVM PSO-GSA Combine	Thyroid	64,50	1	99,793	**99,378***
	Kanker Payudara	96,41	1	99,675	**95,725***
	Kanker Serviks	100	1	92,348	**69,119***

*** : rata-rata akurasi Total 10 Fold**

Tabel 4.34 menunjukkan bahwa akurasi total 10 fold yang tertinggi pada data ketiga data yaitu menggunakan metode LS-SVM PSO-GSA Combine. Akurasi total untuk mendiagnosis penyakit tiroid sebesar 99,77%, akurasi total untuk mendiagnosis penyakit kanker payudara sebesar 100% dan akurasi total untuk mendiagnosis penyakit kanker serviks sebesar 60,02%.

Setalah dilakukan pengklasifikasian dengan metode LS-SVM Sebelum dilakukan *imbalanced* dan setelah dilakukan imbalanced baik menggunakan optimasi PSO-GSA maupun dengan trial error, nilai rata-rata tertinggi pada setiap metode terangkum pada Tabel 4.35 sampai dengan Tabel 4.41.

Tabel 4.35 Rangkuman nilai rata-rata Akurasi tertinggi Pada Training q=10

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	**100*** (C=50,100) (σ=1)	99,885 (C=100) (σ=1)	**100*** (C=100) (σ=1)	99,885 (C=50) (σ=1)	**100*** (C=50,100) (σ=1)	99,793 (C=100) (σ=1)	**100*** (C=100) (σ=1)	99,793 (C=50) (σ=1)
2	95,068 (C=50) (σ=1)	95,220 (C=50) (σ=1)	99,350 (C=1) (σ=1)	99,696 (C=50) (σ=1)	94,944 (C=50) (σ=1)	95,058 (C=50) (σ=1)	99,351 (C=1) (σ=1)	**99,675*** (C=50) (σ=1)
3	88,679 (C=100) (σ=1)	93,381 (C=100) (σ=1)	96,093 (C=100) (σ=1)	**97,091*** (C=100) (σ=1)	90,721 (C=100) (σ=1)	93,902 (C=100) (σ=1)	93,348 (C=100) (σ=1)	92,348 (C=100) (σ=1)

Ket : *) Rata-rata Akurasi Training Tertinggi

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Tabel 4.35 merupakan rangkuman hasil nilai rata-rata akurasi tertinggi training pada semua metode. Pada data thyroid (1), rata-rata akurasi *training* tertinggi sebesar 100%. Pada data kanker payudara (2), rata-rata akurasi training tertinggi dihasilkan oleh metode *Combine* LS-SVM sebesar 99,675%. Pada data kanker serviks (3), rata-rata akurasi training tertinggi dihasilkan oleh metode Tomek Links LS-SVM PSO-GSA sebesar 97,091%

Untuk validasi model klasifikasi digunakan data testing dengan (σ) dan (C) yang telah diperoleh selama proses *training*. Rangkuman nilai rata-rata Akurasi klasifikasi tertinggi pada data *testing* di semua metode yang dicobakan dapat dilihat pada Tabel 4.36.

Tabel 4.36 Rangkuman nilai rata-rata Akurasi Total Pada Testing (q=10 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	93,333 (C=100) (σ=10)	99,375 (C=100) (σ=1)	93,000 (C=1) (σ=1)	99,375 (C=100) (σ=1)	96,774 (C=80,50) (σ=2,99)	**99,378*** (C=76,75) (σ=1)	95,109 (C=61,05) (σ=1)	**99,378*** (C=64,50) (σ=1)
2	89,236 (C=50) (σ=20)	91,238 (C=1) (σ=1)	95,474 (C=50,100) (σ=10,20)	93,793 (C=50) (σ=1)	89,620 (C=44,14) (σ=2,36)	92,432 (C=63,60) (σ=2,85)	95,666 (C=73,03) (σ=2,84)	**95,725*** (C=96,41) (σ=1)
3	45,097 (C=1) (σ=1)	69,012 (C=100) (σ=1)	56,167 (C=1) (σ=20)	68,312 (C=100) (σ=1)	39,919 (C=100) (σ=1)	59,554 (C=100) (σ=1)	44,672 (C=100) (σ=1	**69,119*** (C=100) (σ=1)

Ket : *) Rata-rata Akurasi Tertinggi

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Tabel 4.37 Rangkuman nilai rata-rata *Sensitivity* Pada Testing (q=10 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	86,572 (C=100) (σ=10)	97,216 (C=100) (σ=1)	87,109 (C=1) (σ=1)	97,284 (C=100) (σ=1)	83,575 (C=48,58) (σ=1,27)	**98,234*** (C=76,40) (σ=1,01)	82,567 (C=57,56) (σ=1,93)	**98,234*** (C=69,50) (σ=1)
2	89,665 (C=50) (σ=20)	91,675 (C=1) (σ=1)	82,817 (C=50,100) (σ=10,20)	93,245 (C=50) (σ=1)	83,535 (C=11 (σ=1,61)	90,654 (C=71,29) (σ=5,31)	89,998 (C=33,68 (σ=2,82)	**95,830*** (C=48,98 (σ=1)
3	47,101 (C=1) (σ=1)	60,102 (C=100) (σ=1)	56,020 (C=1) (σ=20)	60,543 (C=100) (σ=1)	38,254 (C=100) (σ=1)	59,920 (C=100) (σ=1)	56,256 (C=99,89) (σ=1)	**62,564*** (C=100) (σ=1)

Ket : *) Rata-rata Sensitivity Tertinggi

Tabel 4.38 Rangkuman nilai rata-rata *Specificity* Pada Testing (q=10 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	92,215 (C=100) (σ=10)	97,980 (C=100) (σ=1)	92,001 (C=1) (σ=1)	95,101 (C=100) (σ=1)	89,207 (C=80,50) (σ=2,99)	98,846 (C=76,75) (σ=1)	91,325 (C=61,05) (σ=1)	**98,612*** (C=64,50) (σ=1)
2	95,876 (C=50) (σ=20)	94,781 (C=1) (σ=1)	96,828 (C=50,100) (σ=10,20)	94,876 (C=50) (σ=1)	90,106 (C=44,14) (σ=2,36)	93,654 (C=63,60) (σ=2,85)	94,624 (C=73,03) (σ=2,84)	**96,210*** (C=96,41) (σ=1)
3	77,423 (C=1) (σ=1)	64,231 (C=100) (σ=1)	83,624 (C=1) (σ=20)	85,120 (C=100) (σ=1)	75,721 (C=100) (σ=1)	65,123 (C=100) (σ=1)	84,203 (C=100) (σ=1)	**85,100*** (C=100) (σ=1)

Ket : *) Rata-rata Specificity Tertinggi

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Tabel 4.39 Rangkuman nilai rata-rata P*recision* Pada Testing (q=10 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	86,572 (C=100) (σ=10)	97,216 (C=100) (σ=1)	87,109 (C=1) (σ=1)	97,284 (C=100) (σ=1)	83,575 (C=48,58) (σ=1,27)	**98,234*** (C=76,40) (σ=1,01)	82,567 (C=57,56) (σ=1,93)	**98,234*** (C=69,50) (σ=1)
2	89,665 (C=50) (σ=20)	91,675 (C=1) (σ=1)	82,817 (C=50,100) (σ=10,20)	93,245 (C=50) (σ=1)	83,535 (C=11 (σ=1,61)	90,654 (C=71,29) (σ=5,31)	89,998 (C=33,68 (σ=2,82)	**95,830*** (C=48,98 (σ=1)
3	47,101 (C=1) (σ=1)	60,102 (C=100) (σ=1)	56,020 (C=1) (σ=20)	60,543 (C=100) (σ=1)	38,254 (C=100) (σ=1)	59,920 (C=100) (σ=1)	56,256 (C=99,89) (σ=1)	**62,564*** (C=100) (σ=1)

Ket : *) Rata-rata Precision Tertinggi

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Tabel 4.40 Rangkuman nilai rata-rata Fmeasure Pada Testing (q=10 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	86,572 (C=100) (σ=10)	97,216 (C=100) (σ=1)	87,109 (C=1) (σ=1)	97,284 (C=100) (σ=1)	83,575 (C=48,58) (σ=1,27)	**98,234*** (C=76,40) (σ=1,01)	82,567 (C=57,56) (σ=1,93)	**98,234*** (C=69,50) (σ=1)
2	89,665 (C=50) (σ=20)	91,675 (C=1) (σ=1)	82,817 (C=50,100) (σ=10,20)	93,245 (C=50) (σ=1)	83,535 (C=11 (σ=1,61)	90,654 (C=71,29) (σ=5,31)	89,998 (C=33,68 (σ=2,82)	**95,830*** (C=48,98 (σ=1)
3	47,101 (C=1) (σ=1)	60,102 (C=100) (σ=1)	56,020 (C=1) (σ=20)	60,543 (C=100) (σ=1)	38,254 (C=100) (σ=1)	59,920 (C=100) (σ=1)	56,256 (C=99,89) (σ=1)	**62,564*** (C=100) (σ=1)

Ket : *) Rata-rata Fmeasurey Tertinggi

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Tabel 4.41 Rangkuman nilai rata-rata Gmean Pada Testing (q=10 Fold)

Data	Metode							
	M1	M2	M3	M4	M5	M6	M7	M8
1	89,213 (C=100) (σ=10)	97,624 (C=100) (σ=1)	89,210 (C=1) (σ=1)	97,754 (C=100) (σ=1)	87,278 (C=80,50) (σ=2,99)	**98,237*** (C=76,75) (σ=1)	87,084 (C=61,05) (σ=1)	**98,237*** (C=64,50) (σ=1)
2	92,112 (C=50) (σ=20)	91,577 (C=1) (σ=1)	94,791 (C=50,100) (σ=10,20)	94,116 (C=50) (σ=1)	87,203 (C=44,14) (σ=2,36)	93,278 (C=63,60) (σ=2,85)	94,434 (C=73,03) (σ=2,84)	**95,896*** (C=96,41) (σ=1)
3	59,772 (C=1) (σ=1)	71,233 (C=100) (σ=1)	67,876 (C=1) (σ=20)	71,636 (C=100) (σ=1)	51,222 (C=100) (σ=1)	73,321 (C=100) (σ=1)	69,567 (C=100) (σ=1)	**73,124*** (C=100) (σ=1)

Ket : *) Rata-rata Gmean Tertinggi

Tabel 4.36 sampai dengan Tabel 4.41 merupakan rangkuman hasil nilai rata-rata akurasi, *Sensitivity, Specificity, Precision, Fmeasure, Gmean* tertinggi testing pada semua metode. Hasil menunjukkan bahwa metode Combine LS-SVM PSO-GSA merupakan metode terbaik atau unggul di semua data percobaan baik diukur performansi dari akurasi total, *Sensitivity, Specificity, Precision, Fmeasure* dan *Gmean.*

Berdasarkan Hasil dari Tabel 4.36 sampai dengan Tabel 4.41, maka telah diketahui metode yang menghasilkan akurasi tertinggi pada setiap data percobaan. Selanjutnya akan dituliskan model persamaan LS-SVM pada setiap data berdasarkan metode terbaiknya. Model disusun berdasarkan nilai rata-rata akurasi tertinggi pada Training.

Pada data thyroid memiliki 3 kelas maka akan terbentuk tiga model persamaan. Persamaan model *Combine* LS-SVM PSO-GSA *multiclass* pada data thyroid dapat ditulis sebagai berikut:

i. Untuk kelas 1 dan kelas 2 (C= 64,50 dan σ=1)

Diketahui :

$x_i=[x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}]$,i=1,2,...,415

$x_j=[x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}]$, j=1,2,...,415

Diperoleh :

b = -0,0253

α_i berukuran 415×1 ($\alpha_1 = 0,1718$; $\alpha_2 = 0,2441; ...; \alpha_{415} = 0$)

Maka

$\widehat{f}_1(\mathbf{x}) = (\widehat{\mathbf{w}}^T \text{x+b})$

dimana $\widehat{\mathbf{w}} = \sum_{i=1}^{415}\sum_{j=1}^{415} \alpha_i y_i K\left(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j}\right)$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\|\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right\|^2}{2\sigma^2}\right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 1 dan kelas 2) adalah

$$\widehat{f}_1(\mathbf{x}) = \sum_{i=1}^{415}\sum_{j=1}^{415} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right) - 0,0253$$

ii. Untuk kelas 1 dan kelas 3 (C= 64,50 dan σ=1)

Diketahui :

$x_i=[x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}]$,i=1,2,...,415

$x_j=[x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}]$, j=1,2,...,415

Diperoleh :

b = -0,3932

α_i berukuran 483×1 ($\alpha_1 = -0,1374$; $\alpha_2 = 0,1585; ...; \alpha_{415} = -0,3455$)

Maka

$\hat{f}_2(\mathbf{x}) = (\hat{\mathbf{w}}^T x+b)$

dimana $\hat{\mathbf{w}}= \sum_{i=1}^{415}\sum_{j=1}^{415} \alpha_i y_i K(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j})$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\|\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right\|^2}{2\sigma^2}\right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 1 dan kelas 3) adalah

$$\hat{f}_2(\mathbf{x}) = \sum_{i=1}^{415}\sum_{j=1}^{415} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right) - 0,3932$$

iii. Untuk kelas 1 dan kelas 3 (C= 69,50 dan σ=1)

Diketahui :

$x_i=[x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}]$,i=1,2,...,415

$x_j=[x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}]$, j=1,2,...,415

Diperoleh :

b = -0,11888

α_i berukuran 415×1 ($\alpha_1 = 0$; $\alpha_2 = 0; ...; \alpha_{415} = 0,0186$)

Maka

$\hat{f}_3(\mathbf{x}) = (\hat{\mathbf{w}}^T x+b)$

dimana $\hat{\mathbf{w}}= \sum_{i=1}^{415}\sum_{j=1}^{415} \alpha_i y_i K(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j})$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\| \mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right\|^2}{2\sigma^2} \right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 2 dan kelas 3) adalah

$$\widehat{f}_3(\mathbf{x}) = \sum_{i=1}^{415} \sum_{j=1}^{415} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right) - 0{,}1188$$

Pada data Kanker Payudara memiliki 3 kelas maka akan terbentuk tiga model persamaan. Persamaan model *Combine* LS-SVM PSO-GSA *multiclass* pada data kanker payudara dapat ditulis sebagai berikut:

i. Untuk kelas 1 dan kelas 2 (C= 96,41 dan σ=1)

Diketahui :

$x_i = [x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}, x_{6i}]$,i=1,2,...,298

$x_j = [x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}, x_{6i}]$, j=1,2,...,298

Diperoleh :

b = 0,3512

α_i berukuran 298×1 ($\alpha_1 = -0{,}6345$; $\alpha_2 = 0; ...; \alpha_{298} = -1{,}3234$)

Maka

$\widehat{f}_1(\mathbf{x}) = (\widehat{\mathbf{w}}^T \mathrm{x} + \mathrm{b})$

dimana $\widehat{\mathbf{w}} = \sum_{i=1}^{298} \sum_{j=1}^{298} \alpha_i y_i K\left(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j} \right)$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\| \mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right\|^2}{2\sigma^2} \right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 1 dan kelas 2) adalah

$$\widehat{f}_1(\mathbf{x}) = \sum_{i=1}^{298} \sum_{j=1}^{298} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j} \right)^2}{2(1)^2} \right) \right) + 0{,}3512$$

ii. Untuk kelas 1 dan kelas 3 (C= 96,41 dan σ=1)

Diketahui :

$x_i = [x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}, x_{6i}]$,i=1,2,...,298

$x_j = [x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}, x_{6i}]$, j=1,2,...,298

Diperoleh :

b = -0,5976

α_i berukuran 298×1 ($\alpha_1 = 0$; $\alpha_2 = 2{,}123; ...; \alpha_{298} = 0$)

Maka

$\hat{f}_1(\mathbf{x}) = (\hat{\mathbf{w}}^T \text{x+b})$

dimana $\hat{\mathbf{w}} = \sum_{i=1}^{298}\sum_{j=1}^{298} \alpha_i y_i K\left(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j}\right)$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\|\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right\|^2}{2\sigma^2}\right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 1 dan kelas 3) adalah

$$\hat{f}_1(\mathbf{x}) = \sum_{i=1}^{298}\sum_{j=1}^{298} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right) - 0{,}5976$$

iii. Untuk kelas 2 dan kelas 3 (C= 96,41 dan σ=1)

Diketahui :

$\text{x}_i = [\text{x}_{1i}, \text{x}_{2i}, \text{x}_{3i}, \text{x}_{4i}, \text{x}_{5i}, \text{x}_{6i}]$,i=1,2,...,298

$\text{x}_j = [\text{x}_{1j}, \text{x}_{2j}, \text{x}_{3j}, \text{x}_{4j}, \text{x}_{5j}, \text{x}_{6i}]$, j=1,2,...,298

Diperoleh :

b = -0,3213

α_i berukuran 298×1 ($\alpha_1 = 0{,}4321$; $\alpha_2 = 4{,}210; ...; \alpha_{292} = 1{,}4324$)

Maka

$\hat{f}_1(\mathbf{x}) = (\hat{\mathbf{w}}^T \text{x+b})$

dimana $\hat{\mathbf{w}} = \sum_{i=1}^{298}\sum_{j=1}^{298} \alpha_i y_i K\left(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j}\right)$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\|\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right\|^2}{2\sigma^2}\right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 2 dan kelas 3) adalah

$$\hat{f}_1(\mathbf{x}) = \sum_{i=1}^{298}\sum_{j=1}^{298} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right) - 0{,}3213$$

Pada data Kanker serviks memiliki 5 kelas maka akan terbentuk 10 model persamaan. Persamaan model *Combine* LS-SVM PSO-GSA *multiclass* pada data kanker serviks dapat ditulis sebagai berikut:

Untuk kelas 2 dan kelas 3 (C= 100 dan σ=1) adalah

Diketahui :

$x_i=[x_{1i}, x_{2i}, x_{3i}, x_{4i}, x_{5i}, x_{6i}, x_{7i}]$,i=1,2,...,559

$x_j=[x_{1j}, x_{2j}, x_{3j}, x_{4j}, x_{5j}, x_{6i}, x_{7i}]$, j=1,2,...,559

Diperoleh :

$b = -0,0432$

α_i berukuran 559×1 ($\alpha_1 = -0,23295$; $\alpha_2 = 0;...;\alpha_{559} = -1,3267$)

Maka

$\widehat{f}_1(\mathbf{x}) = (\widehat{\mathbf{w}}^T x+b)$

dimana $\widehat{\mathbf{w}}= \sum_{i=1}^{559}\sum_{j=1}^{559} \alpha_i y_i K\left(\mathbf{x}_{trainining\ i}, \mathbf{x}_{training\ j}\right)$, dan

$$K(\mathbf{X}_{trainining\ i}, \mathbf{X}_{training\ j}) = \exp\left(-\frac{\left\|\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right\|^2}{2\sigma^2}\right) = \left(\exp\left(-\frac{\left(\mathbf{X}_{trainining\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right)$$

Sehingga persamaan model untuk fungsi pemisah pertama (kelas 1 dan kelas 2) adalah

$$\widehat{f}_1(\mathbf{x}) = \sum_{i=1}^{559}\sum_{j=1}^{559} \alpha_i y_i \left(\exp\left(-\frac{\left(\mathbf{X}_{training\ i} - \mathbf{X}_{training\ j}\right)^2}{2(1)^2}\right)\right) - 0,0432$$

4.2.6 Perbandingan Kebaikan Metode dengan Uji Friedman Untuk 10 Fold

Secara deskripsi, metode yang unggul atau metode yang terbaik adalah Combine LS-SVM PSO-GSA. Untuk membuktikan mana metode yang terbaik pada semua data percobaan digunakan pengujian Friedman. Pengujian Friedman pada q-fold (q=10) adalah sebagai berikut.

(i) Hipotesis

H_0 : $R_1 = R_2 = R_3 = R_4$ (tidak ada metode yang berbeda)

H_1 : minimal ada satu dari R_j berbeda atau tidak sama

(ii) Hipotesis

H_0 : $R_5 = R_6 = R_7 = R_8$ (tidak ada metode yang berbeda)

H_1 : minimal ada satu dari R_j berbeda atau tidak sama

(iii) Hipotesis

H_0 : $R_1 = R_2 = ... = R_8$ (tidak ada metode yang berbeda)

H_1 : minimal ada satu dari R_j berbeda atau tidak sama

Tingkat Signifikan : ($\alpha = 5\%$)

Daerah Kritis : Tolak H_0 jika $\chi^2_{hitung} > \chi^2_{df,\alpha}$ atau p-Value< α

Tabel 4.42 Uji Perbandingan Kebaikan Metode Klasifikasi LS-SVM (q=10) berdasarkan nilai rata-rata akurasi total dengan Uji Friedman

Metode	Chis-Square	Df	p-Value	Keputusan
M1 s/d M4	4,241	3	0,237	Gagal Tolak H_0
M5 s/d M8	5,800	3	0,122	Gagal Tolak H_0
M1 s/d M8	9,454	7	0,222	Gagal Tolak H_0

Ket : $\chi^2_{3;0,05} = 7,815$; $\chi^2_{7;0,05} = 14,067$

Tabel 4.43 Uji Perbandingan Kebaikan Metode Klasifikasi LS-SVM (q=10) berdasarkan nilai rata-rata Sensitivity dengan Uji Friedman

Metode	Chis-Square	Df	p-Value	Keputusan
M1 s/d M4	8,200	3	0,042*	Tolak H_0
M5 s/d M8	6,517	3	0,089	Gagal Tolak H_0
M1 s/d M8	15,645	7	0,029*	Tolak H_0

Ket : *) Minimal ada salah satu metode yang berbeda pada tingkat signifikan α=5%

$\chi^2_{3;0,05} = 7,815$; $\chi^2_{7;0,05} = 14,067$

Tabel 4.44 Uji Perbandingan Kebaikan Metode Klasifikasi LS-SVM (q=10) berdasarkan nilai rata-rata *G-mean* dengan Uji Friedman

Metode	Chis-Square	Df	p-Value	Keputusan
M1 s/d M4	3,800	3	0,284	Gagal Tolak H_0
M5 s/d M8	5,690	3	0,128	Gagal Tolak H_0
M1 s/d M8	13,861	7	0,054	Gagal Tolak H_0

Ket : $\chi^2_{3;0,05} = 7,815$; $\chi^2_{7;0,05} = 14,067$

M1 : LS-SVM Original
M2 : LS-SVM SMOTE
M3 : LS-SVM Tomek Links
M4 : LS-SVM Combine Sampling
M5 : LS-SVM PSO-GSA Original
M6 : LS-SVM PSO-GSA SMOTE
M7 : LS-SVM PSO-GSA Tomek Links
M8 : LS-SVM PSO-GSA Combine Sampling

Berdasarkan Tabel 4.38, dengan menggunakan tingkat signfikansi ($\alpha = 5\%$) disimpulkan bahwa pada metode LS-SVM tanpa penanganan *imbalanced* dan menggunakan *imbalanced* data (M1 s/d M4), pada LS-SVM optimasi PSO-GSA tanpa menggunakan penanganan *imbalanced* dan menggunakan *imbalanced* data (M5 s/d M8) serta pada kedelapan metode yang dicobakan menghasilkan akurasi testing (validasi model) yang sama atau tidak ada metode yang terbaik pada tingkat signifikan 5%.

Berdasarkan Tabel 4.39, dengan menggunakan tingkat signfikansi ($\alpha = 5\%$) disimpulkan bahwa pada metode LS-SVM tanpa penanganan *imbalanced* dan menggunakan *imbalanced* data (M1 s/d M4), LS-SVM PSO-GSA tanpa menggunakan penanganan *imbalanced* dan menggunakan *imbalanced* data (M5 s/d M8) serta pada kedelapan metode yang dicobakan (M1 s/d M8) menghasilkan akurasi *sensitivity* (validasi model) yang berbeda pada tingkat signifikan 5%. Untuk mengetahui dari kedelapan metode tersebut,mana metode yang berbeda maka dilakukan uji pembandingan berganda.

Hipotesis :

H_0 : $R_j = R_{j^*}$ (tidak terdapat perbedaan efek perlakuan j dengan j*)

H_1 : $R_j \neq R_{j^*}$ (terdapat perbedaan efek perlakuan j dengan j*) ; j=1,2,…,8

Daerah Kritis : Tolak H_0 jika $\left|R_j - R_{j^*}\right| > Z_{\{1-(\alpha/k(k-1)\}}\sqrt{\frac{bk(k+1)}{6}}$

dimana :

$$Z_{\left\{1-\left(0,05/8(8-1)\right\}}\sqrt{\frac{3(8)(8+1)}{6}} = 2,36(6) = 14,16$$

Tabel 4.45 Pembandingan Berganda Berdasarkan nilai *Sensitivity*

Jumlah ranking	Metode	M1	M2	M3	M4	M5	M6	M7	M8
8	M1	0							
16	M2	-8	0						
8	M3	0	8	0					
19	M4	-11	-3	-11	0				
5	M5	3	11	3	14	0			
16,5	M6	-8,5	-0,5	-8,5	2,5	-11,5	0		
12	M7	-4	4	-4	**7**	-7	4,5	0	
23,5	M8	**-15,5***	-7,5	**-15,5***	-4,5	**-18,5***	-7	-11,5	0

Ket : *) Tolak H_0 pada tingkat signifikan 5%

Berdasarkan Tabel 4.41, dengan menggunakan tingkat signfikansi ($\alpha = 5\%$) disimpulkan bahwa metode yang berbeda adalah metode 5 dan metode 8, metode 1 dan metode 8, metode 3 dan metode 8. Jumlah ranking metode 5 (R_5) lebih kecil dari Jumlah ranking metode 8 (R_8) , Jumlah ranking metode 1 (R_1) lebih kecil dari Jumlah ranking metode 8 (R_8) dan Jumlah ranking metode 3 (R_3) lebih kecil dari Jumlah ranking metode 8 (R_8) , maka dapat disimpulkan bahwa metode yang terbaik dalam mengukur performansi (akurasi) kelas positif (minor) adalah dengan menggunakan metode 4 (Combine LS-SVM PSO-GSA).

Berdasarkan Tabel 4.40, dengan menggunakan tingkat signfikansi ($\alpha = 5\%$) disimpulkan bahwa pada metode LS-SVM tanpa penanganan *imbalanced* dan menggunakan *imbalanced* data (M1 s/d M4), LS-SVM PSO-GSA tanpa menggunakan penanganan *imbalanced* dan menggunakan *imbalanced* data (M5 s/d M8) serta pada kedelapan metode yang dicobakan (M1 s/d M8) menghasilkan *G-mean* testing (validasi model) yang sama atau tidak ada metode yang terbaik pada tingkat signifikan 5%.

4.2.7 Uji Perbandingan Kebaikan q-*Fold Cross Validation*

Pada penelitian ini menggunakan pembagian data *training* dan *testing* dengan *Q-Fold Cross validation* (q=5) dan (q=10). Berdasarkan hasil rata-rata

akurasi validasi model (testing) tertinggi yang terangkum pada Tabel 4.19 dan 4.37, menunjukan bahwa dengan menggunakan Q–*Fold crossvalidation* (q=10) terlihat menghasilkan akurasi yang lebih tinggi dibandingkan dengan menggunakan (q=5).

Untuk membuktikan mana q-*fold cross validation* terbaik pada semua data percobaan digunakan pengujian *mann whitney*. Kelompok data q-*fold* (q=5) dan q-fold (q=10) adalah independen. Pengujian *mann whitney* pada data thyroid, data kanker payudara dan kanker serviks adalah sebagai berikut.

Hipotesis

H_0 : $R_1 = R_2$ (jumlah ranking q=5 sama dengan q=10)

H_1 : $R_1 \neq R_2$ (jumlah ranking q=5 tidak sama dengan q=10)

Tingkat Signifikan : ($\alpha = 5\%$)

Daerah Kritis : Tolak H_0 jika $Z_{hitung} > Z_{\alpha/2}$ atau $Z_{hitung} < -Z_{\alpha/2}$ atau p-Value< α

Tabel 4.46 Uji Mann Whitney Perbandingan Kebaikan q-*Fold Cross Validation*

Data	W	Ties	Z_{hitung}	p-Value	Keputusan
Rata-rata Akurasi Total					
Thyroid	16	8	-1,707	0,088	Gagal Tolak H_0
Kanker Payudara	21	0	-1,156	0,279	Gagal Tolak H_0
Kanker Serviks	26	0	-0,630	0,574	Gagal Tolak H_0
Rata-rata *Sensitivity*					
Thyroid	30	4	-0,211	0,833	Gagal Tolak H_0
Kanker Payudara	32	0	0,000	1	Gagal Tolak H_0
Kanker Serviks	29	0	-0,315	0,798	Gagal Tolak H_0
Rata-rata *G-mean*					
Thyroid	32	4	0,000	1	Gagal Tolak H_0
Kanker Payudara	21	0	-1,156	0,279	Gagal Tolak H_0
Kanker Serviks	24,5	0	-0,788	0,442	Gagal Tolak H_0

Keterangan : $Z_{0,05/2} = 1,96$

Berdasarkan Tabel 4.42, diperoleh nilai $Z > -Z_{\alpha/2}$ atau p-Value > α sehingga dapat disimpulkan bahwa dengan menggunakan q-*fold cross validation* (q=10) menghasilkan nilai akurasi total, *sensitivity*, *G-mean* yang sama dengan menggunakan (q=5) pada semua data percobaan atau dapat dikatakan bahwa klasifikasi dengan menggunakan q-fold (q=10) sama baiknya dengan menggunakan q-fold (q=5).

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan dari hasil dan pembahasan penelitian ini adalah

1. Tahap Combine Sampling yang digunakan dalam penelitian ini yaitu melakukan proses *preprocessing imbalanced* data dengan SMOTE kemudian dilanjutkan dengan proses *preprocessing imbalanced data* Tomek Links. Desain Algoritma Combine Sampling yaitu menentukan kelas minor dan mayor, menghitung jarak Euclidean, menentukan replikasi dari data minor, menentukan data yang akan direplikasi pada kelas data minor, menghitung sintetis data kemudian mengidentifikasi data kelas mayor dan minor yang dekat dengan data kelas mayor ataupun minor, mengidentifikasi apakah kasus Tomek Links atau tidak dan akhirnya jika ada kasus Tomek Links maka data mayor dieliminasi.
2. Metode yang terbaik untuk kasus klasifikasi *imbalanced* dalam memprediksi status pasien penderita Thyroid, kanker payudara dan kanker serviks adalah metode *combine Sampling Least Square Support Vector Machine PSO-GSA*. Klasifikasi dengan menggunakan Q-Fold (q=5) dan (q=10) menghasilkan performansi yang sama dalam hal akurasi Total, *Sensitivity* dan *G-mean*.

Pada dasarnya baik metode LS-SVM maupun LS-SVM PSO-GSA mengharuskan peneliti untuk lebih tepat dalam menentukan parameter (*trial and error* nilai C dan σ untuk LS-SVM; range nilai C dan σ, inisialisasi parameter PSO dan GSA untuk LS-SVM PSO-GSA).

5.2 Saran

Berdasarkan kesimpulan yang diperoleh, saran yang dapat dipertimbangkan untuk penelitian selanjutnya adalah

1. Melakukan simulasi untuk setiap kategori *imbalanced data* yaitu kategori *imbalanced* tingkat tinggi, sedang, dan rendah.

2. Menggunakan *Stratified Cross Validation* dalam membagi data training dan data testing agar proporsi kelas dapat seimbang..
3. Penelitian selanjutnya disarankan untuk mencoba menggunakan optimasi yang lain, misalnya Genetika algoritma.

LAMPIRAN

Lampiran 1 Data Thyroid

No	X1	X2	X3	X4	X5	Y
1	107	10.1	2.2	0.9	2.7	1
2	113	9.9	3.1	2	5.9	1
3	127	12.9	2.4	1.4	0.6	1
4	109	5.3	1.6	1.4	1.5	1
5	105	7.3	1.5	1.5	-0.1	1
6	105	6.1	2.1	1.4	7	1
7	110	10.4	1.6	1.6	2.7	1
8	114	9.9	2.4	1.5	5.7	1
9	106	9.4	2.2	1.5	0	1
10	107	13	1.1	0.9	3.1	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
151	139	16.4	3.8	1.1	-0.2	2
152	111	16	2.1	0.9	-0.1	2
153	113	17.2	1.8	1	0	2
154	65	25.3	5.8	1.3	0.2	2
155	88	24.1	5.5	0.8	0.1	2
156	134	16.4	4.8	0.6	0.1	2
157	110	20.3	3.7	0.6	0.2	2
158	67	23.3	7.4	1.8	-0.6	2
159	95	11.1	2.7	1.6	-0.3	2
160	89	14.3	4.1	0.5	0.2	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮
184	125	2.3	0.9	16.5	9.5	3
185	120	6.8	2.1	10.4	38.6	3
186	108	3.5	0.6	1.7	1.4	3
205	103	5.1	1.4	1.2	5	3
206	97	4.7	1.1	2.1	12.6	3
207	102	5.3	1.4	1.3	6.7	3

Keterangan :

X1 = Persentase Hasil Uji Asam T3 (T3 Resin)
X2 = Total Serum Thyroxin (T4)
X3 = Total Serum Triidothyronine
X4 = Hormon Basal Thyroid Stimulating (TSH)
X5 = Perbedaan Maximal Absolute pada nilai TSH setelah disuntik
Y = Kondisi Thyroid
(1= Normal, "2"=Hyperthyroidism, "3"=Hypothyroidism)

Lampiran 2 Data Kanker Payudara

No	X1	X2	X3	X4	X5	X6	Y
1	1	0	0	0	1	2	2
2	1	0	0	1	0	2	2
3	3	0	0	1	1	2	3
4	3	3	0	1	1	2	3
5	1	1	0	1	0	2	2
6	1	2	1	1	0	1	3
7	2	1	0	1	1	2	3
8	1	0	0	1	0	2	2
9	1	2	0	1	1	2	3
10	1	0	0	1	1	2	2
11	1	0	0	1	1	2	2
12	3	3	1	1	1	1	3
13	1	1	1	1	0	2	2
14	1	2	1	1	0	2	3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
166	1	2	0	1	0	1	3
167	1	1	0	1	0	1	2
168	0	0	0	0	0	2	1
169	1	2	1	1	0	1	3
170	2	0	1	1	0	1	2
171	1	0	1	1	0	1	2
172	3	1	1	1	0	2	3
173	1	0	0	1	0	1	2
174	1	0	0	1	0	1	2
175	1	2	0	1	0	1	2
176	1	1	0	1	1	2	3
177	1	1	0	1	0	2	2
178	0	2	1	1	0	1	3

Keterangan :

X1 = Ukuran Tumor
X2 = Nodus
X3 = Kemoterapi
X4 = Tingkat Keganasan
X5 = Letak Kanker
X6 = Usia Pasien
Y = Jenis Stadium Pasien

Lampiran 3 Data Kanker Serviks

no	X1	X2	X3	X4	X5	X6	X7	Y
1	38	2	14	26	1	1	1	1
2	44	1	15	24	1	2	1	2
3	37	1	13	29	2	1	1	2
4	30	1	13	26	1	1	1	3
5	46	1	13	45	1	2	1	2
6	40	2	13	22	1	1	1	2
7	42	2	13	24	2	1	1	2
8	43	2	15	20	2	2	2	4
9	38	1	17	27	1	1	2	2
10	48	2	11	19	2	1	2	1
11	36	2	13	22	2	2	1	2
12	35	2	12	27	1	1	1	2
13	54	2	13	42	2	1	1	1
14	41	1	12	35	1	1	2	1
15	30	1	13	17	2	1	2	1
16	40	2	12	28	1	1	1	2
17	54	1	14	25	1	1	2	2
18	37	1	11	23	2	1	1	2
19	38	1	12	29	1	1	2	1
20	52	2	12	25	1	1	1	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
780	57	1	13	18	2	1	1	2
781	47	1	12	22	1	1	1	2
782	37	2	14	25	1	1	1	4
783	49	2	11	12	2	1	1	1
784	40	1	17	20	2	1	2	2
785	34	2	13	25	2	1	1	2
786	49	1	11	24	2	1	2	1
787	39	2	15	22	2	1	1	4
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
792	41	2	10	35	1	1	2	1
793	45	1	14	26	1	1	1	4
794	46	1	13	26	1	1	2	4

Keterangan :
X1 = Usia Pasien Saat Melakukan Pemeriksaan
X2 = Penggunaan Kontrasepsi
X3 = Usia Mestruasi Pertama Kali
X4 = Usia Pertama Kali Melahirkan
X5 = Paritas
X6 = Siklus Menstruasi
X7 = Riwayat Keguguran
Y = Jenis Stadium Pasien

Lampiran 4 Syntak Macro Minitab Deteksi Outlier Multivariate

```
macro
outlier obs  y.1-y.p
mconstant i n p  df
mcolumn d x.1-x.p y.1-y.p dd pi f_value tt obs p1  sig_f
mmatrix s sinv ma mb mc md
let n=count(y.1)
cova y.1-y.p s
invert s sinv
do i=1:p
  let x.i=y.i-mean(y.i)
enddo
do i=1:n
  copy x.1-x.p ma;
    use i.
  transpose ma mb
  multiply ma sinv mc
  multiply mc mb md
  copy md tt
  let d(i)=tt(1)
enddo
let f_value=((n-p-1)*n*d)/(p*(n-1)**2-n*p*d)
let df=n-p-1
cdf f_value p1;
  f p df.
let sig_f=1-p1
print obs d f_value sig_f
endmacro
```

Hasil Deteksi Outlier

Row	C1	d	Fhitung	P-value
156	65	41.9035	12.8575	0
167	80	34.6241	10.1904	0
193	118	33.3762	9.7549	0
195	119	84.8196	34.7391	0
196	123	47.2119	14.9502	0
199	121	41.3608	12.6509	0
208	112	41.8087	12.8213	0
204	136	28.4968	8.1085	0.000004
210	119	24.1982	6.7286	0.000041
170	68	24.063	6.6862	0.000044
187	120	20.2491	5.5152	0.000306
159	67	15.0856	4.0016	0.003771
165	89	14.2987	3.7778	0.005458

Lampiran 5 FUNGSI PROGRAM SMOTE Untuk Kasus Kanker Serviks

```
function [original_featuresTot original_markTot] = SMOTENew(original_features,
original_mark)
Knearest3                                                = 3;
Knearest4                                                = 6;
th3                                                      = 0.5;
th4                                                      = 0.5;

% SMOTE_Class 3
ind3  = find(original_mark == 3);
Xi3   = original_features(ind3,:);
T3    = Xi3';
P3    = T3;
% SMOTE_Class 4
ind4  = find(original_mark == 4);
Xi4   = original_features(ind4,:);
T4    = Xi4';
P4    = T4;
% SMOTE_Class 5
ind51   = find(original_mark == 5);
Xi51    = original_features(ind51,:);
T51     = Xi51';
P51     = T51;
% Menghitung xnearest 3
Xknn3 = nearestneighbour(T3, P3, 'NumberOfNeighbours',Knearest3);
% SMOTE 3 (Xreplika = (1-alpha)*Xi + alpha*Xknn)
Xknn3   = Xknn3'; % --> Xknn hasil dari nearest
[r3 c3] = size(Xknn3);
S3      = [];
for i3=1:r3
    for j3=1:c3
        index3  = Xknn3(i3,j3);
        new_Xi3 = ((1-th3).*Xi3(i3,:)) + (th3.*Xi3(index3,:));
        S3      = [S3;new_Xi3];
    end
end
% Menghitung xnearest 4
Xknn4 = nearestneighbour(T4, P4, 'NumberOfNeighbours',Knearest4);
% SMOTE 4 (Xreplika = (1-alpha)*Xi + alpha*Xknn)
Xknn4   = Xknn4'; % --> Xknn hasil dari nearest
[r4 c4] = size(Xknn4);
S4      = [];
for i4=1:r4
    for j4=1:c4
        index4  = Xknn4(i4,j4);
        new_Xi4 = ((1-th4).*Xi4(i4,:)) + (th4.*Xi4(index4,:));
        S4      = [S4;new_Xi4];
    end
end
% Menghitung xnearest 51 % Replikasi pertama untuk Class 5
Knearest51 = 6;
th51       = 0.9;
Xknn51     = nearestneighbour(T51, P51, 'NumberOfNeighbours',Knearest51);

% SMOTE 5 (Xreplika = (1-alpha)*Xi + alpha*Xknn)
Xknn51    = Xknn51'; % --> Xknn hasil dari nearest
[r51 c51] = size(Xknn51);
S51       = [];
for i51=1:r51
    for j51=1:c51
        index51  = Xknn51(i51,j51);
        new_Xi51 = ((1-th51).*Xi51(i51,:)) + (th51.*Xi51(index51,:));
        S51      = [S51;new_Xi51];
    end
end
```

```
% Data baru setelah replika dari metode SMOTE
original_featuresTot1 = [original_features;S3;S4;S51]; % Data Input Awal di tambah
hasil replika

% Data Class 3
[r3 c3]          = size(S3);
mark3            = ones(r3,1)*3;

% Data Class 4
[r4 c4]          = size(S4);
mark4            = ones(r4,1)*4;

% Data Class 5
[r51 c51]          = size(S51);
mark51             = ones(r51,1)*5;
original_markTot1  = [original_mark;mark3;mark4;mark51];

% SMOTE_Class 52
ind52   = find(original_markTot1 == 5);
Xi52    = original_featuresTot1(ind52,:);
T52     = Xi52';
P52     = T52;

% Menghitung xnearest 52 % Replikasi kedua untuk Class 5
Knearest52 = 6;
th52       = 0.9;
Xknn52     = nearestneighbour(T52, P52, 'NumberOfNeighbours',Knearest52);
% SMOTE 5 (Xreplika = (1-alpha)*Xi + alpha*Xknn)
Xknn52    = Xknn52'; % --> Xknn hasil dari nearest
[r52 c52] = size(Xknn52);
S52       = [];
for i52=1:r52
    for j52=1:c52
        index52  = Xknn52(i52,j52);
        new_Xi52 = ((1-th52).*Xi52(i52,:)) + (th52.*Xi52(index52,:));
        S52      = [S52;new_Xi52];
    end
end
% Data baru setelah replika dari metode SMOTE
original_featuresTot = [original_featuresTot1;S52]; % Data Input Awal di tambah
hasil replika
% Data Class 5 kedua
[r52 c52]          = size(S52);
mark52             = ones(r52,1)*5;
original_markTot   = [original_markTot1;mark52];
end
```

Lampiran 6 PROGRAM NEAREST NEIGHBOR

```
function [idx, tri] = nearestneighbour(varargin)
%NEARESTNEIGHBOUR    find nearest neighbours
error(nargchk(1, Inf, nargin, 'struct'));

% Default parameters
userParams.NumberOfNeighbours = []    ; % Finds one
userParams.DelaunayMode       = 'auto'; % {'on', 'off', |'auto'|}
userParams.Triangulation      = []    ;
userParams.Radius             = inf   ;

% Parse inputs
[P, X, fIndexed, userParams] = parseinputs(userParams, varargin{:});

% Special case uses Delaunay triangulation for speed.

% Determine whether to use Delaunay - set fDelaunay true or false
nX  = size(X, 2);
nP  = size(P, 2);
dim = size(X, 1);

switch lower(userParams.DelaunayMode)
   case 'on'
      %TODO Delaunay can't currently be used for finding more than one
      %neighbour
      fDelaunay = userParams.NumberOfNeighbours == 1 && ...
         size(X, 2) > size(X, 1)                 && ...
         ~fIndexed                               && ...
         userParams.Radius == inf;
   case 'off'
      fDelaunay = false;
   case 'auto'
      fDelaunay = userParams.NumberOfNeighbours == 1 && ...
         ~fIndexed                               && ...
         size(X, 2) > size(X, 1)                 && ...
         userParams.Radius == inf                && ...
         ( ~isempty(userParams.Triangulation) || delaunaytest(nX, nP, dim) );
end
% Try doing Delaunay, if fDelaunay.
fDone = false;
if fDelaunay
   tri = userParams.Triangulation;
   if isempty(tri)
      try
         tri   = delaunayn(X');
      catch
         msgId = 'NearestNeighbour:DelaunayFail';
         msg = ['Unable to compute delaunay triangulation, not using it. ',...
            'Set the DelaunayMode parameter to "off"'];
         warning(msgId, msg);
      end
   end
   if ~isempty(tri)
      try
         idx = dsearchn(X', tri, P')';
         fDone = true;
      catch
         warning('NearestNeighbour:DSearchFail', ...
            'dsearchn failed on triangulation, not using Delaunay');
      end
   end
else % if fDelaunay
   tri = [];
end
% If it didn't use Delaunay triangulation, find the neighbours directly by
% finding minimum distances
if ~fDone
   idx = zeros(userParams.NumberOfNeighbours, size(P, 2));
```

```
% Loop through the set of points P, finding the neighbours
    Y = zeros(size(X));
    for iPoint = 1:size(P, 2)
        x = P(:, iPoint);

        % This is the faster than using repmat based techniques such as
        % Y = X - repmat(x, 1, size(X, 2))
        for i = 1:size(Y, 1)
            Y(i, :) = X(i, :) - x(i);
        end

        % Find the closest points, and remove matches beneath a radius
        dSq = sum(abs(Y).^2, 1);
        iRad = find(dSq < userParams.Radius^2);
        if ~fIndexed
            iSorted = iRad(minn(dSq(iRad), userParams.NumberOfNeighbours));
        else
            iSorted = iRad(minn(dSq(iRad), userParams.NumberOfNeighbours + 1));
            iSorted = iSorted(2:end);
        end

        % Remove any bad ones
        idx(1:length(iSorted), iPoint) = iSorted';
    end
    %while ~isempty(idx) && isequal(idx(end, :), zeros(1, size(idx, 2)))
    %    idx(end, :) = [];
    %end
    idx( all(idx == 0, 2), :) = [];
end % if ~fDone
if isvector(idx)
    idx = idx(:)';
end
end % nearestneighbour
%DELAUNAYTEST   Work out whether the combination of dimensions makes
%fastest to use a Delaunay triangulation in conjunction with dsearchn.
%These parameters have been determined empirically on a Pentium M 1.6G /
%WinXP / 512MB / Matlab R14SP3 platform. Their precision is not
%particularly important
function tf = delaunaytest(nx, np, dim)
switch dim
    case 2
        tf = np > min(1.5 * nx, 400);
    case 3
        tf = np > min(4 * nx  , 1200);
    case 4
        tf = np > min(40 * nx , 5000);

        % if the dimension is higher than 4, it is almost invariably better not
        % to try to use the Delaunay triangulation
    otherwise
        tf = false;
end % switch
end % delaunaytest
%MINN   find the n most negative elements in x, and return their indices
%  in ascending order
function I = minn(x, n)

% Make sure n is no larger than length(x)
n = min(n, length(x));

% Sort the first n
[xsn, I] = sort(x(1:n));
% Go through the rest of the entries, and insert them into the sorted block
% if they are negative enough
for i = (n+1):length(x)
    j = n;
    while j > 0 && x(i) < xsn(j)
        j = j - 1;
    end
```

Lampiran7 PROGRAM TOMEK LINKS

```
%-----------------------------------------------------------%
%-----                  Tomek Link                   -----%
%-----------------------------------------------------------%

%-- 1. Load Data --%
%- Note :
%    1. Variabel respon diletakkan di paling kanan
%    2. File disimpan didalam notepad
clear; clc;

%- Pilih data apa yang ingin digunakan (Hilangkan tanda '%')
%Data                                                    =
xlsread('Kanker_Serviks.xls'); % Load data dari Excel
Data                                                    = xlsread('smote'); %
Load data dari Excel

[nData p]=size(Data);

%-- 2. Hitung Frekuensi Variabel Respon
Y=unique(Data(:,p)); Y(:,2)=zeros; nY=length(Y);
for i=1:nY
    temp=0;
    for j=1:nData
        if Data(j,p)==Y(i,1)
            temp=temp+1;
        end;
    end;
    Y(i,2)=temp;
end;

%-- 3. Mengurutkan Variabel Respon
for i=1:(nY-1)
    for j=(i+1):nY
        if Y(j,2)>Y(i,2)
            temp=Y(j,:);
            Y(j,:)=Y(i,:);
            Y(i,:)=temp;
        end;
    end;
end;

%-- 4. Buat Letak Variabel Respon
Letak=zeros(max(Y(:,2)),length(Y));
for i=1:nY
    temp=0;
    for j=1:nData
        if Data(j,p)==Y(i,1)
            temp=temp+1;
            Letak(temp,i)=j;
        end;
    end;
end;

%-- 5. Lakukan Tomek Link
%-- 5.1. Hitung matriks d
d=zeros(nData, nData);
for i=1:nData
    for j=1:nData
        temp=0;
        for k=1:(p-1)
            temp=temp+(Data(i,k)-Data(j,k)).^2;
        end;
        d(j,i)=sqrt(temp);
    end;
end;
```

```
%-- 5.2. Mencari titik yang terdekat
ihilang=0; iHasil=0;
for j=2:nY
    for k=1:length(Letak(:,1))
        l=1;
        for l=1:length(Letak(:,j))
            if Letak(l,j)==0
                l=l;
            else
                distance=0;
                for i=1:nData
                    if (Letak(k,1)==i) || (Letak(l,j)==i)
                        distance(i)=9999;
                    else
                        temp1=0; temp2=0;
                        for j1=1:(p-1)
                            temp1=temp1+(Data(Letak(k,1),j1)-Data(i,j1)).^2;
                            temp2=temp2+(Data(Letak(l,j),j1)-Data(i,j1)).^2;
                        end;
                        distance(i)=temp1+temp2;
                    end;
                end;

                %- 5.2.a Mencari titik lain yang dekat
                dekat=0;
                for i1=1:nData
                    if distance(i1)==min(distance)
                        dekat=i1;
                    end;
                end;
                base=d(Letak(k,1),Letak(l,j));
                cek1=d(Letak(k,1),dekat);
                cek2=d(Letak(l,j),dekat);

                %- 5.2.b Membuat list kasus Tomek Link
                iHasil=iHasil+1;
                CekHasil(iHasil,:)=[Letak(k,1) Letak(l,j) dekat base cek1 cek2];
                if (cek1>=base) && (cek2>=base)
                    ihilang=ihilang+1;
                    hilang(ihilang)=Letak(k,1);
                end;
            end;
        end;
    end;
end;

%-- 6. Output Data Baru
hilang=unique(hilang);
for i=1:nData
    for j=1:length(hilang)
        if i==hilang(j)
            LetakDataBaru(i,j)=0;
        else
            LetakDataBaru(i,j)=i;
        end;
    end;
end;

for i=1:nData
    temp=unique(LetakDataBaru(i,:));
    if length(temp)>1
        LetakAkhir(i)=0;
    else
        LetakAkhir(i)=temp(1);
    end;
end;
LetakAkhir=unique(round(LetakAkhir));
LetakAkhir=LetakAkhir(2:length(LetakAkhir));
DataBaru=Data(LetakAkhir,:);
hilang
```

Lampiran 8 PROGRAM FUNGSI LS-SVM

```
function [yp,alpha,b,gam,sig2,model] = lssvm(x,y,type,varargin)

if isempty(varargin)
    kernel = 'RBF_kernel';
else
    kernel = varargin{1};
end

if type(1)=='f'
    perffun = 'mse';
elseif type(1)=='c'
    perffun = 'misclass';
else
    error('Type not supported. Choose ''f'' or ''c''')
end

n = size(x,1);
if n <= 300
    optfun = 'leaveoneoutlssvm';
    optargs = {perffun};
else
    optfun = 'crossvalidatelssvm';
    optargs = {10,perffun};
end

model = initlssvm(x,y,type,[],[],kernel);
model = tunelssvm(model,'simplex',optfun,optargs);
model = trainlssvm(model);

if size(x,2) <= 2
    plotlssvm(model);
end


% first output
yp = simlssvm(model,x);

% second output
alpha = model.alpha;

% third output
b = model.b;

% fourth and fifth output
gam = model.gam; sig2 = model.kernel_pars;
```

Lampiran 9 PROGRAM FUNGSI TRAINLSSVM

```
function [model,b,X,Y]  = trainlssvm(model,X,Y)

% initialise the model 'model'
%
if (iscell(model)),
    model = initlssvm(model{:});
end

%
% given X and Y?
%
%model = codelssvm(model);
eval('model = changelssvm(model,''xtrain'',X);',';');
eval('model = changelssvm(model,''ytrain'',Y);',';');
eval('model = changelssvm(model,''selector'',1:size(X,1));',';');

% no training needed if status = 'trained'
%
if model.status(1) == 't',
    if (nargout>1),
        % [alpha,b]
        X = model.xtrain;
        Y = model.ytrain;
        b = model.b;
        model = model.alpha;
    end
    return
end


%
% control of the inputs
%
if ~((strcmp(model.kernel_type,'RBF_kernel') && length(model.kernel_pars)>=1)
||...
        (strcmp(model.kernel_type,'lin_kernel') &&
length(model.kernel_pars)>=0) ||...
        (strcmp(model.kernel_type,'MLP_kernel') &&
length(model.kernel_pars)>=2) ||...
        (strcmp(model.kernel_type,'poly_kernel')&&
length(model.kernel_pars)>=1)),

%
eval('feval(model.kernel_type,model.xtrain(1,:),model.xtrain(2,:),model.kernel
_pars);model.implementation=''MATLAB'';',...
 %       'error(''The kernel type is not valid or to few arguments'');');
elseif (model.steps<=0),
    error('steps must be larger then 0');
elseif (model.gam<=0),
    error('gamma must be larger then 0');
    % elseif (model.kernel_pars<=0),
    %   error('sig2 must be larger then 0');
elseif or(model.x_dim<=0, model.y_dim<=0),
    error('dimension of datapoints must be larger than 0');
end

%
% coding if needed
%
if model.code(1) == 'c', % changed
    model = codelssvm(model);
end

%
% preprocess
%
eval('if model.prestatus(1)==''c'', changed=1; else
changed=0;end;','changed=0;');
if model.preprocess(1) =='p' && changed,
    model = prelssvm(model);
elseif model.preprocess(1) =='o' && changed
```

Lampiran 8 PROGRAM FUNGSI SIMLSSVM

```
function [Y,Yl,model] = simlssvm(model,Xt,A3,A4,A5)

if iscell(model),
    iscell_model = 1;
    model = initlssvm(model{:});
    if iscell(Xt),
        model.alpha = Xt{1};
        model.b = Xt{2};
        model.status = 'trained';
        eval('Xt = A3;',' ');
    end
    eval('nb_to_sim = A4;','nb_to_sim = size(Xt,1)-model.x_delays;');
    Yt = [];
else
    iscell_model = 0;
    if nargin>3,
        Yt = A3;
        eval('nb_to_sim = A4;','nb_to_sim = size(Xt,1)-model.x_delays;');
    else
        eval('nb_to_sim = A3;','nb_to_sim = size(Xt,1)-model.x_delays;');
        Yt =[];
    end
end

eval('Xt;','error(''Test data Xtest undefined...'');');

%
% check dimensions
%
if size(Xt,2)~=model.x_dim,
    error('dimensions of new datapoints Xt not equal to trainingsset...');
end
if ~isempty(Yt) && size(Yt,2)~=model.y_dim,
    error('dimensions of new targetpoints Yt not equal to trainingsset...');
end


%
% preprocessing ...
%
if model.preprocess(1)=='p',
    [Xt,Yt] = prelssvm(model,Xt,Yt);
end

%
% train if status is not  'trained'
%
if model.status(1)~='t', % not 'trained'
    warning('Model is not trained --> training now...')
    model = trainlssvm(model);
end

%
% if dimension of output >1
%
if model.y_dim>1,
    if length(model.kernel_type)>1 || size(model.kernel_pars,2)>1 ||
size(model.gam,2)==model.y_dim,
        %disp('multi dimensional output...');
        fprintf('m');
        [Y Yl] = simmultidimoutput(model,Xt,Yt,nb_to_sim);
        if iscell_model, model = Yl; end
        return
    end
end

%
% set parameters: how much points to evaluate and to simulate
%
if (model.type(1)=='c'),
    nb_sim = nb_to_sim;
    y+=[];
```

Lampiran 10 PROGRAM LS-SVM SMOTE OAO Untuk Kasus Kanker Serviks

```
clear all
clc
close all
tic
Data                                                =
xlsread('Kanker_Serviks.xls'); % Load data dari Excel
DataInput                                           = Data(:,1:7);
DataTarget                                          = Data(:,8);

% Kelas SMOTE
[StadIAsli]                                         = find(DataTarget ==1);
Stadium_I_Asli                                      = length(StadIAsli)
[StadIIAsli]                                        = find(DataTarget ==2);
Stadium_II_Asli                                     = length(StadIIAsli)
[StadIIIAsli]                                       = find(DataTarget ==3);
Stadium_III_Asli                                    = length(StadIIIAsli)
[StadIVAsli]                                        = find(DataTarget ==4);
Stadium_IV_Asli                                     = length(StadIVAsli)
[StadVAsli]                                         = find(DataTarget ==5);
Stadium_V_Asli                                      = length(StadVAsli)

% Program SMOTE
[original_featuresTot original_markTot]             =
SMOTENew(DataInput,DataTarget);
% Kelas SMOTE
[StadISMOTE]                                        = find(original_markTot
==1);
Stadium_I_SMOTE                                     = length(StadISMOTE)
Percent_I                                           =
(Stadium_I_SMOTE/length(original_markTot))*100
[StadIISMOTE]                                       = find(original_markTot
==2);
Stadium_II_SMOTE                                    = length(StadIISMOTE)
Percent_II                                          =
(Stadium_II_SMOTE/length(original_markTot))*100
[StadIIISMOTE]                                      = find(original_markTot
==3);
Stadium_III_SMOTE                                   = length(StadIIISMOTE)
Percent_III                                         =
(Stadium_III_SMOTE/length(original_markTot))*100
[StadIVSMOTE]                                       = find(original_markTot
==4);
Stadium_IV_SMOTE                                    = length(StadIVSMOTE)
Percent_IV                                          =
(Stadium_IV_SMOTE/length(original_markTot))*100
[StadVSMOTE]                                        = find(original_markTot
==5);
Stadium_V_SMOTE                                     = length(StadVSMOTE)
Percent_V                                           =
(Stadium_V_SMOTE/length(original_markTot))*100

 % LS-SVM Classifier
type                                                = 'classifier';
PanjangXfeature                                     =
length(original_featuresTot(:,1));
PanjangXmark                                        =
length(original_markTot(:,1));
```

```
% Validasi data

% Pemilihan data Training dan Testing
Kode_Test  = input('Pilih data testing, pilih salah satu dari data testing dengan
memasukkan angka 1...10: ');
fprintf ('\n')
switch Kode_Test
    case 1
         % Data Testing
         Xtest   = [original_featuresTot(1:round(PanjangXfeature/5),1:7)];
         Ytest   = [original_markTot(1:round(PanjangXmark/5),1)];
         % Data Training
         Xtrain  =
[original_featuresTot((round(PanjangXfeature/5)+1):PanjangXfeature,1:7 )];
         Ytrain  = [original_markTot((round(PanjangXmark/5)+1):PanjangXmark,1)];
    case 2
         % Data Testing
         Xtest   =
[original_featuresTot((round(PanjangXfeature/5)+1):(round(PanjangXfeature/5)*2),1:7)
];
         Ytest   =
[original_markTot((round(PanjangXmark/5)+1):(round(PanjangXmark/5)*2),1)];
         % Data Training
         % Part 1
         Xtrain1 = [original_featuresTot(1:round(PanjangXfeature/5),1:7)];
         Ytrain1 = [original_markTot(1:round(PanjangXmark/5),1)];
         % Part 2
         Xtrain2 =
[original_featuresTot(((round(PanjangXfeature/5)*2)+1):PanjangXfeature,1:7)];
         Ytrain2 = [original_markTot(((round(PanjangXmark/5)*2)+1):PanjangXmark,1)];
         % Total Part 1 dan Part 2
         Xtrain  = [Xtrain1;Xtrain2];
         Ytrain  = [Ytrain1;Ytrain2];
    case 3
         % Data Testing
         Xtest   =
[original_featuresTot(((round(PanjangXfeature/5)*2)+1):(round(PanjangXfeature/5)*3),
1:7)];
         Ytest   =
[original_markTot(((round(PanjangXmark/5)*2)+1):(round(PanjangXmark/5)*3),1)];
         % Data Training
         % Part 1
         Xtrain1 = [original_featuresTot(1:(round(PanjangXfeature/5)*2),1:7)];
         Ytrain1 = [original_markTot(1:(round(PanjangXmark/5)*2),1)];
         % Part 2
         Xtrain2 =
[original_featuresTot(((round(PanjangXfeature/5)*3)+1):PanjangXfeature,1:7)];
         Ytrain2 = [original_markTot(((round(PanjangXmark/5)*3)+1):PanjangXmark,1)];
         % Total Part 1 dan Part 2
         Xtrain  = [Xtrain1;Xtrain2];
         Ytrain  = [Ytrain1;Ytrain2];
    case 4
         % Data Testing
         Xtest   =
[original_featuresTot(((round(PanjangXfeature/5)*3)+1):(round(PanjangXfeature/5)*4),
1:7)];
         Ytest   =
[original_markTot(((round(PanjangXmark/5)*3)+1):(round(PanjangXmark/5)*4),1)];
         % Data Training
         % Part 1
         Xtrain1 = [original_featuresTot(1:(round(PanjangXfeature/5)*3),1:7)];
         Ytrain1 = [original_markTot(1:(round(PanjangXmark/5)*3),1)];
         % Part 2
         Xtrain2 =
[original_featuresTot(((round(PanjangXfeature/5)*4)+1):PanjangXfeature,1:7)];
         Ytrain2 = [original_markTot(((round(PanjangXmark/5)*4)+1):PanjangXmark,1)];
         % Total Part 1 dan Part 2
         Xtrain  = [Xtrain1;Xtrain2];
         Ytrain  = [Ytrain1;Ytrain2];
    otherwise
         % Data Training
         Xtest   =
[original_featuresTot(((round(PanjangXfeature/5)*4)+1):PanjangXfeature,1:7)];
         Ytest   = [original_markTot(((round(PanjangXmark/5)*4)+1):PanjangXmark,1)];
         % Data Testing
         Xtrain  = [original_featuresTot(1:(round(PanjangXfeature/5)*4),1:7)];
         Ytrain  = [original_markTot(1:(round(PanjangXmark/5)*4),1)];
end
```

```
%MELAKUKAN TRAINING MENGGUNAKAN LS-SVM
 [YcodeTr, codebookTr, old_codebookTr]                =
code(Ytrain,'code_OneVsOne');
YcodeTr;
[alpha,bX]                                                =
trainlssvm({Xtrain,YcodeTr,type,C,sigma,'RBF_kernel'});
YTr                                                       =
simlssvm({Xtrain,YcodeTr,type,C,sigma,'RBF_kernel'},{alpha,bX},Xtrain)
;
Yth                                                       =
code(YTr,old_codebookTr,[],codebookTr,[]);
% Hasil Prediksi
[StadITr]       = find(Yth==1);
Stadium_I_Tr    = length(StadITr)
[StadIITr]      = find(Yth==2);
Stadium_II_Tr   = length(StadIITr)
[StadIIITr]     = find(Yth==3);
Stadium_III_Tr = length(StadIIITr)
[StadIVTr]      = find(Yth==4);
Stadium_IV_Tr   = length(StadIVTr)
[StadVTr]       = find(Yth==5);
Stadium_V_Tr    = length(StadVTr)

% Jumlah Benar saat training
[PerformansiTr,JumlahBenarTr,LokasiTr]                =
Benarclass(Ytrain,Yth)
% Jumlah misclass saat training
[PerformansiMissTr,JumlahMissTr,LokasiMissTr]         =
misclass(Ytrain,Yth)
%MELAKUKAN TRAINING MENGGUNAKAN LS-SVM
[YcodeTr, codebookTr, old_codebookTr]                 =
code(Ytrain,'code_OneVsOne');
YcodeTr;
[alpha,bX]                                                =
trainlssvm({Xtrain,YcodeTr,type,C,sigma,'RBF_kernel'});
YTr                                                       =
simlssvm({Xtrain,YcodeTr,type,C,sigma,'RBF_kernel'},{alpha,bX},Xtrain)
;
Yth                                                       =
code(YTr,old_codebookTr,[],codebookTr,[]);
% Hasil Prediksi
[StadITr]       = find(Yth==1);
Stadium_I_Tr    = length(StadITr)
[StadIITr]      = find(Yth==2);
Stadium_II_Tr   = length(StadIITr)
[StadIIITr]     = find(Yth==3);
Stadium_III_Tr = length(StadIIITr)
[StadIVTr]      = find(Yth==4);
Stadium_IV_Tr   = length(StadIVTr)
[StadVTr]       = find(Yth==5);
Stadium_V_Tr    = length(StadVTr)

% Jumlah Benar saat training
[PerformansiTr,JumlahBenarTr,LokasiTr]                =
Benarclass(Ytrain,Yth)
% Jumlah misclass saat training
[PerformansiMissTr,JumlahMissTr,LokasiMissTr]         =
misclass(Ytrain,Yth)
```

Lanjutan Lampiran 10 PROGRAM LS-SVM SMOTE OAO Untuk Kasus Kanker Serviks

```
% Testing

YTs                                                    =
simlssvm({Xtrain,YcodeTr,type,C,sigma,'RBF_kernel'},{alpha,bX},Xtest);
Ytst                                                   =
code(YTs,old_codebookTr,[],codebookTr,[]);
% Hasil Prediksi
[StadITst]       = find(Ytst ==1);
Stadium_I_Tst    = length(StadITst)
[StadIITst]      = find(Ytst ==2);
Stadium_II_Tst   = length(StadIITst)
[StadIIITst]     = find(Ytst ==3);
Stadium_III_Tst  = length(StadIIITst)
[StadIVTst]      = find(Ytst ==4);
Stadium_IV_Tst   = length(StadIVTst)
[StadVTst]       = find(Ytst ==5);
Stadium_V_Tst    = length(StadVTst)
% Jumlah Benar saat testing
[PerformansiTst,JumlahBenarTst,LokasiTst]              =
Benarclass(Ytest,Ytst)
% Jumlah misclass saat testing
[PerformansiMissTst,JumlahMissTst,LokasiMissTst]       =
misclass(Ytest,Ytst)
toc
```

Lampiran 11 PROGRAM SMOTE LS-SVM OAO PSO-GSA KANKER SERVIKS

```
clear all
clc
close all
tic
Data                                                    =
xlsread('Kanker_Serviks.xls'); % Load data dari Excel
DataInput                                               = Data(:,1:7);
DataTarget                                              = Data(:,8);

% Kelas SMOTE
[StadIAsli]                                             = find(DataTarget ==1);
Stadium_I_Asli                                          = length(StadIAsli)
[StadIIAsli]                                            = find(DataTarget ==2);
Stadium_II_Asli                                         = length(StadIIAsli)
[StadIIIAsli]                                           = find(DataTarget ==3);
Stadium_III_Asli                                        = length(StadIIIAsli)
[StadIVAsli]                                            = find(DataTarget ==4);
Stadium_IV_Asli                                         = length(StadIVAsli)
[StadVAsli]                                             = find(DataTarget ==5);
Stadium_V_Asli                                          = length(StadVAsli)

% Program SMOTE
[original_featuresTot original_markTot]                 =
SMOTENew(DataInput,DataTarget);
% Kelas SMOTE
[StadISMOTE]                                            = find(original_markTot
==1);
Stadium_I_SMOTE                                         = length(StadISMOTE)
Percent_I                                               =
(Stadium_I_SMOTE/length(original_markTot))*100
[StadIISMOTE]                                           = find(original_markTot
==2);
Stadium_II_SMOTE                                        = length(StadIISMOTE)
Percent_II                                              =
(Stadium_II_SMOTE/length(original_markTot))*100
[StadIIISMOTE]                                          = find(original_markTot
==3);
Stadium_III_SMOTE                                       = length(StadIIISMOTE)
Percent_III                                             =
(Stadium_III_SMOTE/length(original_markTot))*100
[StadIVSMOTE]                                           = find(original_markTot
==4);
Stadium_IV_SMOTE                                        = length(StadIVSMOTE)
Percent_IV                                              =
(Stadium_IV_SMOTE/length(original_markTot))*100
[StadVSMOTE]                                            = find(original_markTot
==5);
Stadium_V_SMOTE                                         = length(StadVSMOTE)
Percent_V                                               =
(Stadium_V_SMOTE/length(original_markTot))*100

 % LS-SVM Classifier
type                                                    = 'classifier';
PanjangXfeature                                         =
length(original_featuresTot(:,1));
PanjangXmark                                            =
length(original_markTot(:,1));
```

Lanjutan Lampiran 11 PROGRAM SMOTE LS-SVM OAO PSO-GSA KANKER SERVIKS

```
% Inisialisasi parameter PSO
iter_max        =  50;
c1i             = 2.5;
c2i             = 0.5;
c1f             = 0.5;
c2f             = 2.5;
w_max           = 0.9;
w_min           = 0.4;
it              = 1;
W               =(w_max-w_min)*((iter_max -it)/iter_max)+w_min;
swarm           =  20;
jum_par         =   2;

% Parameter GSA
G0              = 10;
acceleration    = zeros(swarm,jum_par);
mass(swarm)     = 0;
force           = zeros(swarm,jum_par);
alpha           = 35;
G               = G0*exp(-alpha *it/iter_max); %Equation (4)
% Initialisasi Parameter ;LS-SVM
%              C   Sig
%              C   Sig
%              C   Sig
Konstraint =[ 100  20   % Maximum
              1  1]; % Minimum
for ir=1:swarm
    for is = 1:jum_par
        Xpar(ir,is)    = Konstraint(2,is)+rand*(Konstraint(1,is)-
Konstraint(2,is));
    end
end
Vpar    = rand(swarm,jum_par)*0;
Fitness = zeros(swarm,1);
% Masuk MultiClass LS-SVM
for ix=1:swarm
    for k=1:5
        % Memanggil kFolds
        Kode_Test=k;
        [Ytrain Xtrain Xtest Ytest]=kFolds1(original_featuresTot,
original_markTot, Kode_Test);

        Xpos=[Xpar(ix,1) Xpar(ix,2)];
        % Melakukan Training Menggunakan LS-SVM
        [YcodeTr, codebookTr, old_codebookTr] = code(Ytrain,'code_OneVsOne');
        [alphaX,bX]                           =
trainlssvm({Xtrain,YcodeTr,type,Xpos(1),Xpos(2),'RBF_kernel'});
        YTr                                   =
simlssvm({Xtrain,YcodeTr,type,Xpos(1),Xpos(2),'RBF_kernel'},{alphaX,bX},Xtrai
n);
        Yth                                   =
code(YTr,old_codebookTr,[],codebookTr,[]);

        % Jumlah Benar saat training
        [PerformansiTr,JumlahTr,LokasiTr]     = Benarclass(Ytrain,Yth);
        k_Perf(k)                                 = PerformansiTr(1);
    end;

    Perf_Classification(ix)               = mean(k_Perf);
    Fitness(ix)                           = Perf_Classification(ix);
end

[Fbest(1),C]      = max(Fitness);
Pbest(1,:)        = Xpar(C,:);
[Fgbest,Iterbest] = max(Fbest);
GlobalBest        = Pbest(Iterbest,:);
FglobalBest(1)    = Fgbest;
worst             = min(Fitness);
best              = max(Fitness);
```

Lanjutan Lampiran 11 PROGRAM SMOTE LS-SVM OAO PSO-GSA KANKER SERVIKS

```
% Gravitational Search Algorithm
% Calculate Mass
for ir=1:swarm
    mass(ir)=(Fitness(ir)-0.99*worst)/(Fgbest-worst);
end
for ir=1:swarm
    mass(ir)=mass(ir)*5/sum(mass);
end
%  Force    update
for ir=1:swarm
    for jr=1:jum_par
        for kr=1:swarm
            if(Xpar(kr,jr)~=Xpar(ir,jr))
               % Equation (3)
              force(ir,jr)=force(ir,jr)+
rand()*G*mass(kr)*mass(ir)*(Xpar(kr,jr)-Xpar(ir,jr))/abs(Xpar(kr,jr)-
Xpar(ir,jr));
            end
        end
    end
end
%  Accelations $ Velocities  UPDATE  %
for ir=1:swarm
    for jr=1:jum_par
        if(mass(ir)~=0)
          % Equation (6)
          acceleration(ir,jr)=force(ir,jr)/mass(ir);
        end
    end
end
% update velocity
c1 = (c1f-c1i)*(it/iter_max)+c1i;
c2 = (c2f-c2i)*(it/iter_max)+c2i;
for ir=1:swarm
    for ik=1:jum_par
        Vpar(ir,ik) =
W*Vpar(ir,ik)+c1*rand()*acceleration(ir,ik)+c2*rand()*(GlobalBest(ik)-
Xpar(ir,ik));
    end
end
Xpar        = Xpar+Vpar;
pmr         = 0.2;
ParticleMut = MutasiParticle(Xpar,swarm,pmr);
Xpar        = ParticleMut;
Xpar        = Xpar+Vpar;
for ix=1:swarm
    if  Xpar(ix,1)< Konstraint(2,1)
        Xpar(ix,1)= Konstraint(2,1);
    elseif Xpar(ix,1)> Konstraint(1,1)
           Xpar(ix,1)= Konstraint(1,1);
    end
    if  Xpar(ix,2)< Konstraint(2,2)
        Xpar(ix,2)= Konstraint(2,2);
    elseif Xpar(ix,2)> Konstraint(1,2)
           Xpar(ix,2)= Konstraint(1,2);
    end
end
```

Lanjutan Lampiran 11 PROGRAM SMOTE LS-SVM OAO PSO-GSA KANKER SERVIKS

```
%  Force    update
      for ir=1:swarm
          for jr=1:jum_par
              for kr=1:swarm
                  if(Xpar(kr,jr)~=Xpar(ir,jr))
                  % Equation (3)
                  force(ir,jr)=force(ir,jr)+
rand()*G*mass(kr)*mass(ir)*(Xpar(kr,jr)-Xpar(ir,jr))/abs(Xpar(kr,jr)-
Xpar(ir,jr));
                  end
              end
          end
      end
      %  Accelations $ Velocities  UPDATE  %
      for ir=1:swarm
          for jr=1:jum_par
              if(mass(ir)~=0)
                 % Equation (6)
                 acceleration(ir,jr)=force(ir,jr)/mass(ir);
              end
          end
      end
      %
==========================================================================
      % update velocity
      c1 = (c1f-c1i)*(it/iter_max)+c1i;
      c2 = (c2f-c2i)*(it/iter_max)+c2i;
      for ir=1:swarm
          for ik=1:jum_par
              Vpar(ir,ik) =
W*Vpar(ir,ik)+c1*rand()*acceleration(ir,ik)+c2*rand()*(GlobalBest(ik)-
Xpar(ir,ik));
          end
      end
      Xpar        = Xpar+Vpar;
      pmr         = 0.9;
      ParticleMut = MutasiParticle(Xpar,swarm,pmr);
      Xpar        = ParticleMut;
      Xpar        = Xpar+Vpar;
      for ix=1:swarm
          if  Xpar(ix,1)< Konstraint(2,1)
              Xpar(ix,1)= Konstraint(2,1);
          elseif Xpar(ix,1)> Konstraint(1,1)
                 Xpar(ix,1)= Konstraint(1,1);
          end
          if  Xpar(ix,2)< Konstraint(2,2)
              Xpar(ix,2)= Konstraint(2,2);
          elseif Xpar(ix,2)> Konstraint(1,2)
                 Xpar(ix,2)= Konstraint(1,2);
          end
      end
      plotvector = get(hbestplot,'YData');
      plotvector(it-1) = FglobalBest(it-1);
      set(hbestplot,'YData',plotvector);
      set(htext1,'String',sprintf('Fungsi Objektif: %f', FglobalBest(it-1)));
      hold on;
      drawnow
end
clear Xpos
Xpos  = [GlobalBest(1) GlobalBest(2)] % P1 % x1 terletak pada kolom 1 sebanyak
jumlah particle dalam kolom (Matriks 1x50)
        % P3
for k=1:5
        % Memanggil kFolds
        Kode_Test=k;
        [Ytrain Xtrain Xtest Ytest]=kFolds1(original_featuresTot,
original_markTot, Kode_Test);
%==========================================================================
```

Lanjutan Lampiran 11 PROGRAM SMOTE LS-SVM OAO PSO-GSA KANKER SERVIKS

```
%====================================================================
======
%MELAKUKAN TRAINING MENGGUNAKAN LS-SVM
%====================================================================
======

[YcodeTr, codebookTr, old_codebookTr]             =
code(Ytrain,'code_OneVsOne');
[alphaX,bX]                                       =
trainlssvm({Xtrain,YcodeTr,type,Xpos(1),Xpos(2),'RBF_kernel'});
YTr                                               =
simlssvm({Xtrain,YcodeTr,type,Xpos(1),Xpos(2),'RBF_kernel'},{alphaX,b
X},Xtrain);
Yth                                               =
code(YTr,old_codebookTr,[],codebookTr,[]);
% Hasil Prediksi
[StadITr]       = find(Yth==1);
Stadium_I_Tr    = length(StadITr)
[StadIITr]      = find(Yth==2);
Stadium_II_Tr   = length(StadIITr)
[StadIIITr]     = find(Yth==3);
Stadium_III_Tr = length(StadIIITr)
% Jumlah Benar saat training
[PerformansiTr,JumlahBenarTr,LokasiTr]            =
Benarclass(Ytrain,Yth)
% Jumlah misclass saat training
[PerformansiMissTr,JumlahMissTr,LokasiMissTr]     =
misclass(Ytrain,Yth)
%====================================================================
======
% MELAKUKAN TESTING MENGGUNAKAN LS-SVM
%====================================================================
======

YTs                                               =
simlssvm({Xtrain,YcodeTr,type,Xpos(1),Xpos(2),'RBF_kernel'},{alphaX,b
X},Xtest);
Ytst                                              =
code(YTs,old_codebookTr,[],codebookTr,[]);
% Hasil Prediksi
[StadITst]       = find(Ytst ==1);
Stadium_I_Tst    = length(StadITst)
[StadIITst]      = find(Ytst ==2);
Stadium_II_Tst   = length(StadIITst)
[StadIIITst]     = find(Ytst ==3);
Stadium_III_Tst = length(StadIIITst)
% Jumlah Benar saat testing
[PerformansiTst,JumlahBenarTst,LokasiTst]         =
Benarclass(Ytest,Ytst)
% Jumlah misclass saat training
[PerformansiMissTst,JumlahMissTst,LokasiMissTst]  =
misclass(Ytest,Ytst)
Perf_Tr(k)=PerformansiTr;
Perf_Tst(k)=PerformansiTst;
end;
PerformansiTr=mean(Perf_Tr);
PerformansiTst=mean(Perf_Tst);
toc
```

Lampiran 12 FUNGSI K-FOLD (K=5) SMOTE

```
function [Ytrain Xtrain Xtest Ytest]=kFolds(DataInput, DataTarget, Kode_Test)
PanjangInput  = length (DataInput(:,1));
PanjangTarget = length (DataTarget(:,1));
switch Kode_Test
    case 1
         % Data Testing
         Xtest   = [DataInput(1:round(PanjangInput/5),1:5)];
         Ytest   = [DataTarget(1:round(PanjangTarget/5),1)];
         % Data Training
         Xtrain  = [DataInput((round(PanjangInput/5)+1):PanjangInput,1:5)];
         Ytrain  = [DataTarget((round(PanjangTarget/5)+1):PanjangTarget,1)];
    case 2
         % Data Testing
         Xtest   =
[DataInput((round(PanjangInput/5)+1):(round(PanjangInput/5)*2),1:5)];
         Ytest   =
[DataTarget((round(PanjangTarget/5)+1):(round(PanjangTarget/5)*2),1)];
         % Data Training
         % Part 1
         Xtrain1 = [DataInput(1:round(PanjangInput/5),1:5)];
         Ytrain1 = [DataTarget(1:round(PanjangTarget/5),1)];
         % Part 2
         Xtrain2 = [DataInput(((round(PanjangInput/5)*2)+1):PanjangInput,1:5)];
         Ytrain2 = [DataTarget(((round(PanjangTarget/5)*2)+1):PanjangTarget,1)];
         % Total Part 1 dan Part 2
         Xtrain  = [Xtrain1;Xtrain2];
         Ytrain  = [Ytrain1;Ytrain2];
    case 3
         % Data Testing
         Xtest   =
[DataInput(((round(PanjangInput/5)*2)+1):(round(PanjangInput/5)*3),1:5)];
         Ytest   =
[DataTarget(((round(PanjangTarget/5)*2)+1):(round(PanjangTarget/5)*3),1)];
         % Data Training
         % Part 1
         Xtrain1 = [DataInput(1:(round(PanjangInput/5)*2),1:5)];
         Ytrain1 = [DataTarget(1:(round(PanjangTarget/5)*2),1)];
         % Part 2
         Xtrain2 = [DataInput(((round(PanjangInput/5)*3)+1):PanjangInput,1:5)];
         Ytrain2 = [DataTarget(((round(PanjangTarget/5)*3)+1):PanjangTarget,1)];
         % Total Part 1 dan Part 2
         Xtrain  = [Xtrain1;Xtrain2];
         Ytrain  = [Ytrain1;Ytrain2];
   case 4
         % Data Testing
         Xtest   =
[DataInput(((round(PanjangInput/5)*3)+1):(round(PanjangInput/5)*4),1:5)];
         Ytest   =
[DataTarget(((round(PanjangTarget/5)*3)+1):(round(PanjangTarget/5)*4),1)];
         % Data Training
         % Part 1
         Xtrain1 = [DataInput(1:(round(PanjangInput/5)*3),1:5)];
         Ytrain1 = [DataTarget(1:(round(PanjangTarget/5)*3),1)];
         % Part 2
         Xtrain2 = [DataInput(((round(PanjangInput/5)*4)+1):PanjangInput,1:5)];
         Ytrain2 = [DataTarget(((round(PanjangTarget/5)*4)+1):PanjangTarget,1)];
         % Total Part 1 dan Part 2
         Xtrain  = [Xtrain1;Xtrain2];
         Ytrain  = [Ytrain1;Ytrain2];

 otherwise
         % Data Testing
         Xtest   = [DataInput(((round(PanjangInput/5)*4)+1):PanjangInput,1:5)];
         Ytest   = [DataTarget(((round(PanjangTarget/5)*4)+1):PanjangTarget,1)];
         % Data Training
         % Part 1
         Xtrain  = [DataInput(1:(round(PanjangInput/5)*4),1:5)];
         Ytrain  = [DataTarget(1:(round(PanjangTarget/5)*4),1)];

end
```

Lampiran 13 PROGRAM TOMEK DAN COMBINE LS-SVM OAO PSO-GSA KANKER SERVIKS

```
clear all
clc
close all
tic
Data                                                =
xlsread('tomek_serviks.xls');
Data                                                =
xlsread('combine.xls'); % Load data dari Excel
% Load data dari Excel
DataInput                                           = Data(:,1:7);
DataTarget                                          = Data(:,8);
%Kelas Asli
[KelasIAsli]                                        = find (DataTarget
==1);
Kelas_I_Asli                                        = length(KelasIAsli);
[KelasIIAsli]                                       = find (DataTarget
==2);
Kelas_II_Asli                                       = length(KelasIIAsli);
[KelasIIIAsli]                                     = find (DataTarget ==3);
Kelas_III_Asli                                      = length(KelasIIIAsli);
[KelasIVAsli]                                       = find (DataTarget
==4);
Kelas_IV_Asli                                       = length(KelasIVAsli);
[KelasVAsli]                                        = find (DataTarget
==5);
Kelas_V_Asli                                        = length(KelasVAsli);


% LS-SVM Classifier
type                                                = 'classifier';
PanjangInput                                        = length
(DataInput(:,1));
PanjangTarget                                       = length
(DataTarget(:,1));

% Inisialisasi parameter PSO
iter_max        =  50;
c1i             = 2.5;
c2i             = 0.5;
c1f             = 0.5;
c2f             = 2.5;
w_max           = 0.9;
w_min           = 0.4;
it              = 1;
W               =(w_max-w_min)*((iter_max -it)/iter_max)+w_min;
swarm           =  20;
jum_par         =   2;

% Parameter GSA
G0              = 10;
acceleration    = zeros(swarm,jum_par);
mass(swarm)     = 0;
force           = zeros(swarm,jum_par);
alpha           = 35;
G               = G0*exp(-alpha *it/iter_max); %Equation (4)
% Initialisasi Parameter ;LS-SVM
%               C   Sig
%               C   Sig
%               C   Sig
Konstraint =[ 100  20   % Maximum
              1  1]; % Minimum
```

Lanjutan Lampiran 13 PROGRAM TOMEK DAN COMBINE LS-SVM OAO PSO-GSA KANKER SERVIKS

```
clear all
clc
close all
tic
Data                                                =
xlsread('tomek_serviks.xls');
Data                                                =
xlsread('combine.xls'); % Load data dari Excel
% Load data dari Excel
DataInput                                           = Data(:,1:7);
DataTarget                                          = Data(:,8);
%Kelas Asli
[KelasIAsli]                                        = find (DataTarget
==1);
Kelas_I_Asli                                        = length(KelasIAsli);
[KelasIIAsli]                                       = find (DataTarget
==2);
Kelas_II_Asli                                       = length(KelasIIAsli);
[KelasIIIAsli]                                     = find (DataTarget ==3);
Kelas_III_Asli                                      = length(KelasIIIAsli);
[KelasIVAsli]                                       = find (DataTarget
==4);
Kelas_IV_Asli                                       = length(KelasIVAsli);
[KelasVAsli]                                        = find (DataTarget
==5);
Kelas_V_Asli                                        = length(KelasVAsli);


% LS-SVM Classifier
type                                                = 'classifier';
PanjangInput                                        = length
(DataInput(:,1));
PanjangTarget                                       = length
(DataTarget(:,1));

% Inisialisasi parameter PSO
iter_max        =  50;
c1i             = 2.5;
c2i             = 0.5;
c1f             = 0.5;
c2f             = 2.5;
w_max           = 0.9;
w_min           = 0.4;
it              = 1;
W               =(w_max-w_min)*((iter_max -it)/iter_max)+w_min;
swarm           =  20;
jum_par         =   2;

% Parameter GSA
G0              = 10;
acceleration    = zeros(swarm,jum_par);
mass(swarm)     = 0;
force           = zeros(swarm,jum_par);
alpha           = 35;
G               = G0*exp(-alpha *it/iter_max); %Equation (4)
% Initialisasi Parameter ;LS-SVM
%               C   Sig
%               C   Sig
%               C   Sig
Konstraint =[ 100  20   % Maximum
              1   1]; % Minimum
```

Lanjutan Lampiran 13 PROGRAM TOMEK DAN COMBINE LS-SVM OAO PSO-GSA KANKER SERVIKS

```
%  Accelations $ Velocities  UPDATE  %
for ir=1:swarm
    for jr=1:jum_par
        if(mass(ir)~=0)
          % Equation (6)
          acceleration(ir,jr)=force(ir,jr)/mass(ir);
        end
    end
end
%
=========================================================================
====
% update velocity
c1 = (c1f-c1i)*(it/iter_max)+c1i;
c2 = (c2f-c2i)*(it/iter_max)+c2i;
for ir=1:swarm
    for ik=1:jum_par
        Vpar(ir,ik) =
W*Vpar(ir,ik)+c1*rand()*acceleration(ir,ik)+c2*rand()*(GlobalBest(ik)
-Xpar(ir,ik));
    end
end
Xpar        = Xpar+Vpar;
pmr         = 0.2;
ParticleMut = MutasiParticle(Xpar,swarm,pmr);
Xpar        = ParticleMut;
Xpar        = Xpar+Vpar;
for ix=1:swarm
    if  Xpar(ix,1)< Konstraint(2,1)
        Xpar(ix,1)= Konstraint(2,1);
    elseif Xpar(ix,1)> Konstraint(1,1)
           Xpar(ix,1)= Konstraint(1,1);
    end
    if  Xpar(ix,2)< Konstraint(2,2)
        Xpar(ix,2)= Konstraint(2,2);
    elseif Xpar(ix,2)> Konstraint(1,2)
           Xpar(ix,2)= Konstraint(1,2);
    end
end
%---Buat Grafik
hfig = figure;
hold on
title('Grafik Konvergensi MultiClass LS-SVM Berbasis PSO');
set(hfig, 'position', [50,40,600,300]);
set(hfig, 'DoubleBuffer', 'on');
hbestplot = plot(1:iter_max,zeros(1,iter_max));
htext1 = text(0.6*iter_max,30,sprintf('Fungsi Fitness : %f', 0.0));
xlabel('Iterasi');
ylabel('Fungsi Fitness(%)');
hold off
drawnow;
```

Lanjutan Lampiran 13 PROGRAM TOMEK DAN COMBINE LS-SVM OAO PSO-GSA KANKER SERVIKS

```
while it<=iter_max
      it=it+1;
      G               = G0*exp(-alpha *it/iter_max); %Equation (4)
      for ix=1:swarm;
            for k=1:5
                % Memanggil kFolds
                Kode_Test=k;
                [Ytrain Xtrain Xtest Ytest]=kFolds(DataInput,
DataTarget, Kode_Test);

                Xpos=[Xpar(ix,1) Xpar(ix,2)];
                % Melakukan Training Menggunakan LS-SVM
                [YcodeTr, codebookTr, old_codebookTr] =
code(Ytrain,'code_OneVsOne');
                [alphaX,bX]                              =
trainlssvm({Xtrain,YcodeTr,type,Xpos(1),Xpos(2),'RBF_kernel'});
                YTr                                      =
simlssvm({Xtrain,YcodeTr,type,Xpos(1),Xpos(2),'RBF_kernel'},{alphaX,bX
},Xtrain);
                Yth                                      =
code(YTr,old_codebookTr,[],codebookTr,[]);

                % Jumlah Benar saat training
                [PerformansiTr,JumlahTr,LokasiTr]     =
Benarclass(Ytrain,Yth);
                k_Perf(k)                                    =
PerformansiTr(1);
            end;
            Perf_Classification(ix)               = mean(k_Perf);
            Fitness(ix)                           =
Perf_Classification(ix);
      end
      [Fbest(it),C]     = max(Fitness);
      Pbest(it,:)       = Xpar(C,:);
      [Fgbest,Iterbest] = max(Fbest);
      GlobalBest        = Pbest(Iterbest,:);
      FglobalBest(it)   = Fgbest;
      worst             = min(Fitness);
      best              = max(Fitness);
%
==========================================================================
==
      % Gravitational Search Algorithm
      % Calculate Mass
      for ir=1:swarm
          mass(ir)=(Fitness(ir)-0.99*worst)/(Fgbest-worst);
      end
      for ir=1:swarm
          mass(ir)=mass(ir)*5/sum(mass);
      end
      %  Force    update
      for ir=1:swarm
          for jr=1:jum_par
              for kr=1:swarm
                  if(Xpar(kr,jr)~=Xpar(ir,jr))
                  % Equation (3)
                  force(ir,jr)=force(ir,jr)+
rand()*G*mass(kr)*mass(ir)*(Xpar(kr,jr)-Xpar(ir,jr))/abs(Xpar(kr,jr)-
Xpar(ir,jr));
                  end
              end
          end
```

Lanjutan Lampiran 13 PROGRAM TOMEK DAN COMBINE LS-SVM OAO PSO-GSA KANKER SERVIKS

```
%  Accelations $ Velocities  UPDATE  %
      for ir=1:swarm
          for jr=1:jum_par
              if(mass(ir)~=0)
                 % Equation (6)
                 acceleration(ir,jr)=force(ir,jr)/mass(ir);
              end
          end
      end
      %
=============================================================
===========
      % update velocity
      c1 = (c1f-c1i)*(it/iter_max)+c1i;
      c2 = (c2f-c2i)*(it/iter_max)+c2i;
      for ir=1:swarm
          for ik=1:jum_par
              Vpar(ir,ik) =
W*Vpar(ir,ik)+c1*rand()*acceleration(ir,ik)+c2*rand()*(GlobalB
est(ik)-Xpar(ir,ik));
          end
      end
      Xpar        = Xpar+Vpar;
      pmr         = 0.9;
      ParticleMut = MutasiParticle(Xpar,swarm,pmr);
      Xpar        = ParticleMut;
      Xpar        = Xpar+Vpar;
      for ix=1:swarm
          if  Xpar(ix,1)< Konstraint(2,1)
              Xpar(ix,1)= Konstraint(2,1);
          elseif Xpar(ix,1)> Konstraint(1,1)
                 Xpar(ix,1)= Konstraint(1,1);
          end
          if  Xpar(ix,2)< Konstraint(2,2)
              Xpar(ix,2)= Konstraint(2,2);
          elseif Xpar(ix,2)> Konstraint(1,2)
                 Xpar(ix,2)= Konstraint(1,2);
          end
      end
      plotvector = get(hbestplot,'YData');
      plotvector(it-1) = FglobalBest(it-1);
      set(hbestplot,'YData',plotvector);
      set(htext1,'String',sprintf('Fungsi Objektif: %f',
FglobalBest(it-1)));
      hold on;
      drawnow
end
clear Xpos
Xpos  = [GlobalBest(1) GlobalBest(2)] % P1 % x1 terletak pada
kolom 1 sebanyak jumlah particle dalam kolom (Matriks 1x50)
        % P3
```

```
for k=1:5
        % Memanggil kFolds
        Kode_Test=k;
        [Ytrain Xtrain Xtest Ytest]=kFolds(DataInput, DataTarget,
Kode_Test);
%=========================================================================
======
%MELAKUKAN TRAINING MENGGUNAKAN LS-SVM
%=========================================================================
======

[YcodeTr, codebookTr, old_codebookTr]              =
code(Ytrain,'code_OneVsOne');
[alphaX,bX]                                        =
trainlssvm({Xtrain,YcodeTr,type,Xpos(1),Xpos(2),'RBF_kernel'});
YTr                                                =
simlssvm({Xtrain,YcodeTr,type,Xpos(1),Xpos(2),'RBF_kernel'},{alphaX,b
X},Xtrain);
Yth                                                =
code(YTr,old_codebookTr,[],codebookTr,[]);
% Hasil Prediksi
[StadITr]       = find(Yth==1);
Stadium_I_Tr    = length(StadITr)
[StadIITr]      = find(Yth==2);
Stadium_II_Tr   = length(StadIITr)
[StadIIITr]     = find(Yth==3);
Stadium_III_Tr = length(StadIIITr)
% Jumlah Benar saat training
[PerformansiTr,JumlahBenarTr,LokasiTr]             =
Benarclass(Ytrain,Yth)
% Jumlah misclass saat training
[PerformansiMissTr,JumlahMissTr,LokasiMissTr]      =
misclass(Ytrain,Yth)
%=========================================================================
======
% MELAKUKAN TESTING MENGGUNAKAN LS-SVM
%=========================================================================
======

YTs                                                =
simlssvm({Xtrain,YcodeTr,type,Xpos(1),Xpos(2),'RBF_kernel'},{alphaX,b
X},Xtest);
Ytst                                               =
code(YTs,old_codebookTr,[],codebookTr,[]);
% Hasil Prediksi
[StadITst]       = find(Ytst ==1);
Stadium_I_Tst    = length(StadITst)
[StadIITst]      = find(Ytst ==2);
Stadium_II_Tst   = length(StadIITst)
[StadIIITst]     = find(Ytst ==3);
Stadium_III_Tst = length(StadIIITst)
% Jumlah Benar saat testing
[PerformansiTst,JumlahBenarTst,LokasiTst]          =
Benarclass(Ytest,Ytst)
% Jumlah misclass saat training
[PerformansiMissTst,JumlahMissTst,LokasiMissTst]   =
misclass(Ytest,Ytst)
Perf_Tr(k)=PerformansiTr;
Perf_Tst(k)=PerformansiTst;
end;
PerformansiTr=mean(Perf_Tr);
PerformansiTst=mean(Perf_Tst);
toc
```

Lampiran14 Uji Friedman Q-Fold Cross Validation (q=5)

Akurasi

Test Statistics[a]

N	3
Chi-Square	14.028
df	7
Asymp. Sig.	.051

a. Friedman Test

Sensitivity

Test Statistics[a]

N	3
Chi-Square	18.200
df	7
Asymp. Sig.	.011

a. Friedman Test

Test Statistics[a]

N	3
Chi-Square	7.966
df	3
Asymp. Sig.	.047

a. Friedman Test

Test Statistics[a]

N	3
Chi-Square	6.517
df	3
Asymp. Sig.	.089

a. Friedman Test

G-mean

Test Statistics[a]

N	3
Chi-Square	12.992
df	7
Asymp. Sig.	.072

a. Friedman Test

Lampiran 15 Uji Friedman Q-Fold Cross Validation (q=10)

Akurasi

Test Statistics[a]

N	3
Chi-Square	4.241
df	3
Asymp. Sig.	.237

a. Friedman Test

Test Statistics[a]

N	3
Chi-Square	9.454
df	7
Asymp. Sig.	.222

a. Friedman Test

Test Statistics[a]

N	3
Chi-Square	5.800
df	3
Asymp. Sig.	.122

a. Friedman Test

Sensitivity

Test Statistics[a]

N	3
Chi-Square	15.645
df	7
Asymp. Sig.	.029

a. Friedman Test

G-mean

Test Statistics[a]

N	3
Chi-Square	13.861
df	7
Asymp. Sig.	.054

a. Friedman Test

Lampiran 16 Uji Mann Whitney Untuk Perbandingan Q-Fold

Test Statistics[a]

	serviks	thyroid	payudara
Mann-Whitney U	26.000	16.000	21.000
Wilcoxon W	62.000	52.000	57.000
Z	-.630	-1.707	-1.156
Asymp. Sig. (2-tailed)	.529	.088	.248
Exact Sig. [2*(1-tailed Sig.)]	.574[b]	.105[b]	.279[b]

a. Grouping Variable: VAR00020

b. Not corrected for ties.

Test Statistics[a]

	thy	pyu	servik
Mann-Whitney U	30.000	32.000	29.000
Wilcoxon W	66.000	68.000	65.000
Z	-.211	.000	-.315
Asymp. Sig. (2-tailed)	.833	1.000	.753
Exact Sig. [2*(1-tailed Sig.)]	.878[b]	1.000[b]	.798[b]

a. Grouping Variable: VAR00060

b. Not corrected for ties.

Test Statistics[a]

	thyroid	payudara	serviks
Mann-Whitney U	32.000	21.000	24.500
Wilcoxon W	68.000	57.000	60.500
Z	.000	-1.156	-.788
Asymp. Sig. (2-tailed)	1.000	.248	.431
Exact Sig. [2*(1-tailed Sig.)]	1.000[b]	.279[b]	.442[b]

a. Grouping Variable: VAR00020

b. Not corrected for ties.

Lampiran 17 Akurasi Training LS-SVM Original

	Parameter		Fold					Fold					
	C	σ	1	2	3	4	5	6	7	8	9	10	rata-rata
Thyroid		1	98.925	98.387	98.387	98.387	98.387	98.387	99.462	98.387	98.925	99.471	98.711
		10	94.624	94.624	95.161	94.624	95.161	94.624	94.624	94.624	95.699	96.825	95.059
	1	20	92.473	92.473	92.473	92.473	92.473	92.473	92.473	93.548	93.548	94.709	92.912
		1	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**
		10	100.000	97.849	98.387	97.849	97.849	97.849	98.387	97.312	98.387	98.942	98.281
	50	20	96.774	96.237	97.849	96.237	96.237	96.237	96.774	96.237	97.849	97.884	96.832
		1	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**
		10	98.387	97.849	98.387	98.387	97.849	98.387	98.387	98.387	98.387	98.942	98.335
	100	20	97.312	97.849	98.387	97.312	97.312	97.312	98.387	97.312	98.387	97.884	97.745
Kanker Payudara	1	1	93.750	95.000	95.000	93.750	93.750	94.375	95.625	93.750	93.750	93.827	94.258
		10	88.750	89.375	90.000	88.125	87.500	89.375	88.750	88.125	88.750	90.123	88.887
		20	87.500	88.125	88.750	87.500	87.500	89.375	87.500	86.250	88.125	90.124	88.075
	50	1	**94.375**	**95.625**	**96.250**	**94.375**	**94.375**	**95.625**	**95.625**	**94.375**	**94.375**	**95.679**	**95.068**
		10	93.750	95.000	95.625	93.125	93.750	95.000	95.625	93.750	93.750	95.062	94.444
		20	92.500	94.375	94.375	93.125	93.125	93.750	95.000	92.500	93.125	94.444	93.632
	100	1	**94.375**	**95.625**	**96.250**	**94.375**	**94.375**	**95.625**	**95.625**	**94.375**	**94.375**	**95.679**	**95.068**
		10	93.750	95.000	95.625	93.125	93.750	95.000	95.625	93.750	93.750	95.062	94.444
		20	92.500	94.375	94.375	92.500	93.125	94.375	95.000	92.500	93.125	94.444	93.632
Kanker Serviks	1	1	73.566	73.007	71.329	71.469	73.287	73.007	73.427	71.608	72.448	73.418	72.656
		10	53.007	53.007	53.566	54.965	54.965	53.427	53.007	53.986	53.566	53.305	53.680
		20	48.532	48.252	49.231	50.769	50.210	50.210	50.210	50.490	50.070	49.789	49.776
	50	1	88.532	88.112	86.853	88.112	86.713	86.993	86.434	86.993	87.832	86.498	87.307
		10	62.518	63.497	63.217	62.937	64.755	64.336	63.916	62.937	61.678	62.729	63.252
		20	39.241	58.322	59.301	57.063	58.741	57.203	57.902	56.504	56.364	59.916	56.055
	100	1	90.070	89.511	87.832	89.371	87.552	88.532	87.552	88.532	89.650	88.186	**88.679**
		10	64.755	65.734	64.895	65.455	66.154	66.434	65.874	65.455	64.336	65.401	65.449
		20	59.580	59.301	60.134	58.601	60.559	59.021	58.741	59.720	58.881	61.322	59.586

Lampiran 18 Akurasi Testing LS-SVM Original

	Parameter		Fold					Fold					rata-rata
	C	σ	1	2	3	4	5	6	7	8	9	10	
Thyroid		1	100.000	100.000	90.476	100.000	100.000	100.000	100.000	52.381	80.952	0.000	82.381
		10	100.000	100.000	100.000	100.000	100.000	100.000	100.000	80.952	66.667	66.667	91.429
	1	20	100.000	100.000	100.000	100.000	100.000	100.000	100.000	66.667	61.905	66.667	89.524
		1	95.238	100.000	90.476	100.000	100.000	100.000	95.238	61.905	80.952	5.555	82.936
		10	97.849	100.000	90.476	100.000	100.000	100.000	100.000	95.238	80.952	66.667	93.118
	50	20	100.000	100.000	100.000	100.000	100.000	100.000	100.000	85.714	71.429	66.667	92.381
		1	92.238	100.000	90.476	100.000	100.000	100.000	95.238	61.905	80.952	5.555	82.636
		10	100.000	100.000	90.476	100.000	100.000	100.000	100.000	95.238	80.952	66.667	**93.333**
	100	20	100.000	100.000	95.238	100.000	100.000	100.000	100.000	95.238	71.429	66.667	92.857
Kanker Payudara	1	1	88.889	88.889	83.333	83.333	94.444	77.778	72.222	94.400	100.000	68.750	85.204
		10	94.444	88.889	83.333	88.889	94.444	77.778	88.889	100.000	88.889	68.750	87.431
		20	94.444	88.889	83.333	88.889	94.444	77.778	94.444	100.000	88.889	68.750	87.986
	50	1	**88.889**	**88.889**	**72.222**	**83.333**	**88.889**	**77.778**	**66.667**	**94.444**	**100.000**	**81.250**	84.236
		10	94.444	88.889	83.333	77.778	100.000	83.333	77.778	94.444	94.444	81.250	87.569
		20	94.444	88.889	83.333	83.333	94.444	94.444	72.222	100.000	100.000	81.250	**89.236**
	100	1	**88.889**	**88.889**	**72.222**	**77.778**	**88.889**	**77.778**	**66.667**	**94.444**	**100.000**	**81.250**	83.681
		10	94.444	88.889	83.333	77.778	100.000	83.333	72.222	94.444	94.444	81.250	87.014
		20	94.444	88.889	83.333	77.778	94.444	88.889	72.222	100.000	100.000	81.250	88.125
Kanker Serviks	1	1	44.304	48.101	48.101	49.367	43.038	49.367	44.304	39.241	41.772	43.374	**45.097**
		10	41.772	45.570	49.367	41.722	41.772	44.304	45.570	49.367	39.241	45.783	44.447
		20	43.038	34.177	46.835	37.975	36.709	45.570	49.367	49.367	39.241	46.988	42.927
	50	1	36.709	44.304	39.241	49.367	37.975	44.304	39.241	44.304	25.317	39.759	40.052
		10	43.038	46.835	44.304	44.304	43.038	49.367	44.304	41.722	40.506	49.398	44.682
		20	57.203	44.304	46.835	43.038	40.506	44.304	46.835	41.772	40.506	44.578	44.988
	100	1	32.911	40.506	36.709	48.101	40.506	44.304	41.772	41.772	25.317	40.964	39.286
		10	43.038	46.835	44.304	45.570	43.038	48.101	43.038	40.506	44.304	45.783	44.452
		20	41.722	44.304	48.101	41.772	41.772	44.304	48.101	40.506	41.772	45.783	43.814

Lampiran 19 Akurasi Training LS-SVM SMOTE

	Parameter		Fold					Fold					rata-rata
	C	σ	1	2	3	4	5	6	7	8	9	10	
Thyroid	1	1	99.310	99.540	99.080	99.310	99.080	99.080	99.310	99.310	99.310	99.306	99.264
		10	98.851	99.770	99.080	98.391	98.161	98.391	98.851	97.012	97.931	96.759	98.320
		20	98.161	99.080	98.851	96.782	97.471	97.011	97.471	96.782	95.552	96.759	97.392
	50	1	**99.770**	**100.000**	**99.540**	**99.770**	**100.000**	**99.770**	**99.770**	**99.770**	**99.770**	**99.769**	99.793
		10	99.080	99.310	99.080	99.770	99.770	99.540	99.310	99.770	99.540	99.537	99.471
		20	98.851	99.540	98.851	99.540	99.540	99.540	99.540	99.540	99.540	98.843	99.333
	100	1	**99.770**	**100.000**	**99.770**	**100.000**	**100.000**	**100.000**	**100.000**	**99.770**	**99.770**	**99.769**	**99.885**
		10	99.080	99.310	99.080	99.540	99.540	99.540	99.540	99.770	99.540	99.537	99.448
		20	90.080	99.310	99.080	95.540	99.540	93.310	93.310	99.540	99.540	99.305	96.856
Kanker Payudara	1	1	94.516	95.161	93.871	94.839	94.839	94.516	94.839	94.193	93.871	94.444	94.509
		10	89.677	90.645	89.023	90.000	89.032	90.323	92.581	92.258	90.968	89.869	90.438
		20	89.677	89.355	88.710	88.065	89.032	89.677	92.581	92.258	90.323	89.869	89.955
	50	1	95.161	95.484	94.516	95.484	95.484	95.161	95.807	94.839	94.839	95.425	**95.220**
		10	93.226	93.226	92.903	93.226	93.226	93.548	94.516	93.871	92.903	93.791	93.444
		20	92.903	92.810	92.258	92.903	92.581	92.581	93.871	93.871	92.581	92.484	92.884
	100	1	95.161	95.484	94.516	95.484	94.839	95.161	95.807	94.839	94.839	95.425	95.155
		10	93.226	94.516	92.903	93.872	94.516	93.548	94.516	93.871	93.871	94.444	93.928
		20	93.548	92.581	92.903	93.871	93.226	93.548	93.871	93.871	92.258	93.464	93.314
Kanker Serviks	1	1	81.959	81.057	81.766	82.088	80.863	80.863	80.477	79.188	77.706	77.390	80.336
		10	57.217	58.119	57.281	57.796	57.796	56.766	54.575	54.124	50.773	51.421	55.587
		20	53.866	55.863	54.381	54.961	51.546	51.997	50.322	50.967	47.294	46.418	51.762
	50	1	94.588	93.814	93.557	93.557	92.526	91.881	91.302	91.624	90.722	90.698	92.427
		10	70.361	70.490	69.523	70.490	70.296	69.588	68.621	68.943	65.335	64.987	68.863
		20	65.271	64.626	64.949	64.884	64.369	63.595	62.822	63.402	59.923	58.527	63.237
	100	1	95.103	94.717	94.072	94.717	93.750	92.912	92.848	92.526	91.495	91.667	**93.381**
		10	72.358	72.680	71.585	72.036	71.843	71.263	70.683	71.199	67.590	67.377	70.861
		20	67.784	67.010	66.817	66.688	66.817	66.817	65.400	65.206	61.856	60.917	65.531

Lampiran 20 Akurasi Testing LS-SVM SMOTE

	Parameter		Fold					Fold					rata-rata
	C	σ	1	2	3	4	5	6	7	8	9	10	
Thyroid	1	1	95.833	93.750	95.833	100.000	100.000	100.000	100.000	100.000	100.000	94.118	97.953
		10	100.000	91.667	97.917	100.000	97.917	100.000	100.000	87.500	100.000	78.431	95.343
		20	100.000	91.667	100.000	97.917	95.833	97.917	97.917	87.500	100.000	70.588	93.934
	50	1	95.833	95.833	100.000	100.000	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	99.167
		10	95.833	93.750	95.833	100.000	100.000	100.000	100.000	100.000	100.000	94.118	97.953
		20	95.833	89.583	93.750	100.000	100.000	100.000	100.000	100.000	100.000	88.235	96.740
	100	1	**97.917**	**95.833**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**99.375**
		10	95.833	93.750	95.833	100.000	100.000	100.000	100.000	100.000	100.000	98.039	98.346
		20	95.833	91.667	93.750	100.000	100.000	100.000	100.000	100.000	100.000	92.157	97.341
Kanker Payudara	1	1	94.118	88.235	94.118	85.294	94.118	94.118	82.353	91.177	94.118	94.737	**91.238**
		10	91.177	79.412	97.059	82.353	97.059	91.176	82.353	91.176	85.294	89.474	88.653
		20	91.176	73.529	91.176	73.529	97.059	91.176	82.353	91.176	85.294	92.105	86.857
	50	1	94.118	**85.294**	**94.118**	**88.235**	**88.235**	**94.118**	**85.294**	**91.177**	**97.059**	**92.105**	90.975
		10	94.118	85.294	94.118	79.412	91.177	91.177	82.353	91.177	94.118	94.737	89.768
		20	94.118	79.412	97.059	85.294	91.177	91.177	82.353	91.177	88.235	89.474	88.947
	100	1	94.118	85.294	94.118	88.235	91.177	94.118	85.294	91.177	97.059	92.105	91.269
		10	94.118	91.177	94.118	79.412	88.235	91.177	82.353	91.177	97.059	94.737	90.356
		20	94.118	79.412	97.059	88.235	91.177	91.177	82.353	91.177	88.235	92.105	89.505
Kanker Serviks	1	1	44.767	41.279	45.930	40.116	60.465	59.302	58.721	56.977	98.256	100.000	60.581
		10	27.326	23.837	28.488	18.605	18.605	34.302	33.721	7.558	86.047	78.409	35.690
		20	25.000	20.930	26.163	15.698	36.047	24.419	26.163	2.907	80.233	59.091	31.665
	50	1	45.930	45.349	53.488	47.674	66.279	76.163	72.674	77.907	97.674	100.000	68.314
		10	37.209	40.116	36.047	33.721	47.674	43.023	54.651	50.581	97.093	98.864	53.898
		20	28.488	31.395	30.814	29.070	44.767	41.279	45.349	35.465	94.186	83.523	46.434
	100	1	45.349	48.837	54.070	47.674	65.116	76.163	76.744	78.488	97.674	100.000	**69.012**
		10	38.372	41.279	39.535	33.140	47.674	47.093	55.814	51.744	97.093	100.000	55.174
		20	30.814	33.140	34.884	29.070	44.186	42.442	48.256	41.861	94.186	86.932	48.577

Lampiran 21 Akurasi Training LS-SVM Tomek Links

	Parameter		Fold					Fold					rata-rata
	C	σ	1	2	3	4	5	6	7	8	9	10	
Thyroid	1	1	98.913	98.369	98.913	100.000	98.913	98.913	99.456	98.369	98.913	100.000	99.076
		10	94.565	94.565	95.109	95.109	95.109	95.109	94.565	94.565	94.565	98.889	95.215
		20	92.391	92.391	92.391	92.391	92.391	92.391	92.391	93.478	94.022	98.333	93.257
	50	1	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**
		10	97.826	97.826	98.369	97.826	97.826	97.826	97.826	98.369	98.369	99.444	98.151
		20	97.283	96.739	98.369	97.283	96.739	96.739	97.826	97.826	97.826	98.889	97.552
	100	1	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**
		10	98.369	98.369	98.913	98.913	98.369	98.369	98.913	98.369	98.369	99.444	98.640
		20	97.283	98.367	98.369	97.283	97.826	97.283	98.369	97.283	97.826	98.889	97.878
Kanker Payudara	1	1	97.842	98.561	97.842	97.842	97.842	97.842	98.561	97.842	97.122	97.037	97.833
		10	92.086	92.086	91.367	91.367	90.648	91.367	93.525	91.367	92.086	92.593	91.849
		20	90.648	90.648	90.648	89.928	89.928	90.648	92.086	89.928	91.367	92.593	90.842
	50	1	99.281	99.281	99.281	99.281	99.281	99.281	100.000	99.281	99.281	99.259	**99.350**
		10	97.842	98.561	97.842	97.122	97.842	97.842	99.281	97.842	97.842	97.778	97.979
		20	96.403	97.842	97.122	97.122	97.122	97.122	98.561	97.122	97.122	97.778	97.332
	100	1	99.281	92.806	99.281	99.281	99.281	99.281	100.000	99.281	99.281	99.259	98.703
		10	97.842	98.561	97.842	97.842	97.842	97.842	99.281	97.842	97.842	97.778	98.051
		20	97.122	97.842	97.122	97.122	97.122	97.122	98.561	97.842	97.122	97.778	97.476
Kanker Serviks	1	1	76.978	76.439	75.180	76.079	77.698	76.079	78.058	76.799	76.978	76.882	76.717
		10	59.353	60.252	59.892	59.892	60.612	58.993	59.173	60.612	62.050	60.036	60.086
		20	58.453	58.813	57.734	57.734	59.173	57.374	57.374	58.273	60.072	56.452	58.145
	50	1	92.446	92.446	92.446	92.806	91.187	92.266	91.007	91.727	92.626	91.756	92.071
		10	69.065	69.784	67.986	69.964	70.684	69.425	68.705	70.334	69.784	69.355	69.508
		20	64.029	64.389	63.669	64.029	66.187	65.108	64.389	51.613	65.827	64.158	63.340
	100	1	93.345	94.604	93.166	94.065	92.986	93.525	92.626	92.446	93.345	93.369	**93.348**
		10	71.403	72.302	69.964	72.302	73.022	71.223	71.223	73.201	71.942	72.581	71.916
		20	66.007	65.468	63.669	66.007	67.086	66.547	65.288	67.086	66.547	65.950	65.965

Lampiran 22 Akurasi Testing LS-SVM Tomek Links

	Parameter		Fold										rata-rata
	C	σ	1	2	3	4	5	6	7	8	9	10	
Thyroid		1	100.000	100.000	95.000	100.000	100.000	100.000	100.000	75.000	60.000	100.000	**93.000**
		10	100.000	100.000	100.000	100.000	100.000	100.000	100.000	75.000	70.000	0.000	84.500
	1	20	100.000	100.000	100.000	100.000	100.000	100.000	100.000	75.000	55.000	0.000	83.000
		1	**100.000**	**100.000**	**95.000**	**100.000**	**100.000**	**100.000**	**50.000**	**90.000**	**60.000**	**0.000**	79.500
		10	100.000	100.000	95.000	100.000	100.000	100.000	100.000	95.000	80.000	0.000	87.000
	50	20	100.000	100.000	100.000	100.000	100.000	100.000	100.000	90.000	80.000	0.000	87.000
		1	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**90.000**	**60.000**	**0.000**	85.000
		10	100.000	100.000	95.000	100.000	100.000	100.000	100.000	95.000	85.000	0.000	87.500
	100	20	100.000	100.000	95.000	100.000	100.000	100.000	100.000	85.000	80.000	0.000	86.000
Kanker Payudara	1	1	93.333	93.333	93.333	93.333	93.333	93.333	73.333	100.000	93.333	84.211	91.087
		10	93.333	93.333	93.333	100.000	93.333	93.333	80.000	100.000	86.667	78.947	91.228
		20	93.333	93.333	93.333	100.000	93.333	93.333	80.000	100.000	86.667	78.947	91.228
	50	1	93.333	93.333	100.000	86.667	93.333	100.000	80.000	93.333	93.333	89.474	92.281
		10	100.000	93.333	100.000	93.333	100.000	100.000	80.000	100.000	93.333	94.737	95.474
		20	100.000	93.333	100.000	93.333	100.000	100.000	80.000	100.000	93.333	94.737	95.474
	100	1	93.333	93.333	100.000	86.667	93.333	100.000	80.000	93.333	93.333	89.474	92.281
		10	100.000	93.333	100.000	93.333	100.000	100.000	80.000	100.000	93.333	94.737	**95.474**
		20	100.000	93.333	100.000	93.333	100.000	100.000	80.000	100.000	93.333	94.737	**95.474**
Kanker Serviks	1	1	58.065	54.839	59.677	56.452	41.936	54.839	51.613	46.774	45.161	60.000	52.935
		10	54.839	54.839	62.903	53.226	48.387	61.290	64.516	53.226	43.548	60.000	55.677
		20	54.839	56.452	61.290	53.226	48.387	62.903	66.129	53.226	43.548	61.667	**56.167**
	50	1	38.710	46.774	46.774	51.613	41.936	43.548	40.323	46.774	38.710	48.333	44.349
		10	53.226	54.839	58.065	51.613	40.323	58.065	59.677	46.774	45.161	60.000	52.774
		20	58.065	51.613	62.903	50.000	45.161	56.452	64.516	51.613	43.548	58.333	54.220
	100	1	41.936	51.613	43.548	48.387	43.548	43.548	41.936	40.323	40.323	48.333	44.349
		10	58.065	53.226	58.065	48.387	41.936	53.226	54.840	41.936	46.774	53.333	50.979
		20	56.452	53.226	59.677	50.000	43.5484	59.677	61.290	50.000	43.548	58.333	53.575

Lampiran 23 Akurasi Training LS-SVM Combine Sampling

	Parameter		Fold										
	C	σ	1	2	3	4	5	6	7	8	9	10	**rata-rata**
Thyroid		1	99.310	99.540	99.080	99.310	99.080	99.080	99.310	99.310	99.310	99.306	99.264
		10	98.851	99.770	99.080	98.391	98.161	98.391	98.851	tim	97.931	96.759	98.465
	1	20	98.161	99.080	98.851	96.782	97.471	97.011	97.471	96.782	95.552	96.759	97.392
		1	**99.770**	**100.000**	**99.540**	**99.770**	**100.000**	**99.770**	**99.770**	**99.770**	**99.770**	**99.769**	99.793
		10	99.080	99.310	99.080	99.770	99.770	99.540	99.310	99.770	99.540	99.537	99.471
	50	20	98.851	99.540	98.851	99.540	99.540	99.540	99.540	99.540	99.540	98.843	99.333
		1	**99.770**	**100.000**	**99.770**	**100.000**	**100.000**	**100.000**	**100.000**	**99.770**	**99.770**	**99.769**	**99.885**
		10	99.080	99.310	99.080	99.540	99.540	99.540	99.540	99.770	99.540	99.537	99.448
	100	20	90.080	99.310	99.080	95.540	99.540	93.310	93.310	99.540	99.540	99.305	96.856
Kanker Payudara	1	1	98.099	99.240	97.719	98.099	98.099	97.719	97.719	98.859	97.719	96.935	98.020
		10	89.354	92.015	88.973	89.734	88.973	89.354	89.354	95.057	90.114	92.337	90.526
		20	88.973	90.875	88.973	88.973	89.734	89.354	88.593	95.057	88.973	89.655	89.916
	50	1	100.000	99.620	99.620	99.620	99.620	99.620	99.620	100.000	99.620	99.617	**99.696**
		10	95.818	95.578	95.437	96.578	95.818	95.437	95.437	98.479	95.437	95.402	95.942
		20	95.437	94.677	95.437	95.437	95.437	95.437	95.437	98.479	95.437	93.487	95.470
	100	1	100.000	99.620	99.620	99.620	99.620	99.620	99.620	100.000	99.620	99.617	**99.696**
		10	97.338	98.479	97.338	97.338	97.338	97.338	96.958	98.479	96.958	96.935	97.450
		20	95.437	95.437	95.437	95.437	95.437	95.437	95.437	98.479	95.437	93.870	95.585
Kanker Serviks	1	1	82.540	82.209	82.341	78.704	81.878	80.688	81.151	80.093	78.439	78.704	80.675
		10	56.878	57.407	56.217	57.077	56.085	56.944	56.349	55.688	52.050	51.521	55.622
		20	53.704	54.564	54.233	54.167	52.712	52.712	50.265	51.257	47.156	46.164	51.693
	50	1	95.106	94.511	94.312	94.577	93.651	92.593	92.725	92.593	91.601	91.667	93.333
		10	71.032	71.495	70.437	71.561	71.164	70.238	70.238	70.304	65.807	65.675	69.795
		20	66.667	64.947	65.609	65.939	65.807	65.079	63.426	64.220	60.185	59.061	64.094
	100	1	95.635	95.569	95.841	95.503	94.841	93.915	93.717	93.386	92.659	92.791	**94.386**
		10	73.545	73.611	72.950	74.008	72.818	71.759	72.5529.	72.355	68.651	68.585	72.031
		20	68.717	67.196	67.593	67.593	67.328	67.262	65.873	65.939	61.905	61.310	66.071

Lampiran 24 Akurasi Testing LS-SVM Combine Sampling

	Parameter		Fold					Fold					rata-rata
	C	σ	1	2	3	4	5	6	7	8	9	10	
Thyroid	1	1	95.833	93.750	95.833	100.000	100.000	100.000	100.000	100.000	100.000	94.118	97.953
		10	100.000	91.667	97.917	100.000	97.917	100.000	100.000	87.500	100.000	78.431	95.343
		20	100.000	91.667	100.000	97.917	95.833	97.917	97.917	87.500	100.000	70.588	93.934
	50	1	95.833	95.833	100.000	100.000	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	99.167
		10	95.833	93.750	95.833	100.000	100.000	100.000	100.000	100.000	100.000	94.118	97.953
		20	95.833	89.583	93.750	100.000	100.000	100.000	100.000	100.000	100.000	88.235	96.740
	100	1	**97.917**	**95.833**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**100.000**	**99.375**
		10	95.833	93.750	95.833	100.000	100.000	100.000	100.000	100.000	100.000	98.039	98.346
		20	95.833	91.667	93.750	100.000	100.000	100.000	100.000	100.000	100.000	92.157	97.341
Kanker Payudara	1	1	96.552	86.207	96.552	89.655	93.103	100.000	100.000	68.966	100.000	93.548	92.458
		10	93.103	75.862	96.552	86.207	96.552	93.103	100.000	68.966	96.552	83.871	89.077
		20	89.655	75.862	89.655	86.207	96.552	93.103	100.000	68.966	93.104	74.194	86.730
	50	1	96.552	86.207	93.103	100.000	93.103	96.552	100.000	72.414	100.000	100.000	**93.793**
		10	100.000	79.310	93.103	89.655	96.552	96.552	100.000	68.966	100.000	93.548	91.769
		20	100.000	75.862	96.552	93.103	100.000	96.552	100.000	68.966	100.000	83.871	91.491
	100	1	96.552	86.207	93.103	96.552	93.103	96.552	100.000	72.414	100.000	100.000	93.448
		10	100.000	89.655	93.103	89.655	89.655	96.552	100.000	68.965	100.000	100.000	92.759
		20	100.000	79.310	96.552	93.103	100.000	96.552	100.000	68.965	100.000	87.097	92.158
Kanker Serviks	1	1	45.833	41.071	49.405	100.000	57.143	51.191	53.571	49.405	100.000	100.000	64.762
		10	26.786	23.810	27.976	18.452	40.476	16.071	27.381	16.071	92.262	77.381	36.667
		20	23.810	20.238	25.000	13.691	36.310	36.310	13.095	2.976	79.762	57.143	30.833
	50	1	48.810	48.214	55.357	48.810	61.905	69.048	72.024	76.191	100.000	100.000	68.036
		10	36.310	39.286	38.095	32.143	47.024	35.119	51.191	48.810	100.000	98.810	52.679
		20	28.571	30.357	30.357	29.762	42.262	29.167	41.071	37.500	97.024	84.524	45.060
	100	1	47.024	48.810	56.548	47.619	59.524	71.214	76.191	76.191	100.000	100.000	**68.312**
		10	39.286	41.667	41.667	34.524	47.024	39.881	54.762	51.191	100.000	100.000	55.000
		20	31.548	34.524	33.333	29.762	41.667	31.548	44.048	43.452	97.024	89.881	47.679

Lampiran 25 Confusion Matrix Metode LS-SVM Original 5 Fold

thyroid					
Fold	aktual	prediksi			jumlah
		kelas 1	kelas 2	kelas 3	
1	kelas 1	41	0	0	41
	kelas 2	0	0	0	0
	kelas 3	0	0	0	0
	jumlah	41	0	0	41
2	kelas 1	41	0	0	41
	kelas 2	0	0	0	0
	kelas 3	0	0	0	0
	jumlah	41	0	0	41
3	kelas 1	41	0	0	41
	kelas 2	0	0	0	0
	kelas 3	0	0	0	0
	jumlah	41	0	0	41
4	kelas 1	27	0	0	27
	kelas 2	1	12	1	14
	kelas 3	0	0	0	0
	jumlah	28	12	1	41
5	kelas 1	0	0	0	0
	kelas 2	3	16	0	19
	kelas 3	8	16	0	24
	jumlah	11	32	0	43

Kanker Payudara					
Fold	aktual	prediksi			jumlah
		kelas 1	kelas 2	kelas 3	
1	kelas 1	1	0	0	1
	kelas 2	0	14	2	16
	kelas 3	0	1	18	19
	jumlah	1	15	20	36
2	kelas 1	1	0	0	1
	kelas 2	0	18	1	19
	kelas 3	0	3	13	16
	jumlah	1	21	14	36
3	kelas 1	4	0	0	4
	kelas 2	0	6	2	8
	kelas 3	0	1	23	24
	jumlah	4	7	25	36
4	kelas 1	0	0	1	1
	kelas 2	2	9	0	11
	kelas 3	0	0	24	24
	jumlah	2	9	25	36
5	kelas 1	2	2	0	4
	kelas 2	0	12	1	13
	kelas 3	0	3	14	17
	jumlah	2	17	15	34

HALAMAN INI SENGAJA DIKOSONGKAN

BIOGRAFI PENULIS

Penulis dilahirkan di Surabaya, pada tanggal 9 Februari 1991 sebagai anak terakhir dari tiga bersaudara. Penulis bertempat tinggat di Ambengan Batu 2/41 Surabaya. Selama ini penulis telah menempuh pendidikan formal yaitu TK Dharmawanita Surabaya, SDN Sidotopo Wetan I/255 Surabaya, SLTPN 9 Surabaya dan SMAN 3 Surabaya. Setelah lulus dari SMAN tahun 2009, penulis mengikuti seleksi penerimaan mahasiswa baru di ITS dan diterima di jurusan Diploma III Statistika FMIPA-ITS, terdaftar dengan Nrp 1309.030.054 dan melanjutkan S1 Tahun 2012 dengan terdaftar NRP 1312105027. melanjutkan S2 Tahun 2014 dengan terdaftar NRP 1314201044. Alamat email penulis yaitu hanikhaulasari@gmail.com dan dapat menghubungi penulis di 085731848484.

Surabaya, Februari 2016
hanikhaulasari@gmail.com

DAFTAR PUSTAKA

Abdullah, M., (2013). *Wide area Control System Menggunakan Least Square Support Vector Machine dan Improved Quantum Inspired Evolutionary Algorithm Untuk Meredam Osilasi Pada Sistem Tenaga Listrik Dua Arah*, Tesis, Teknik Elektro, FTI-ITS, Surabaya. Diakses dari http://digilib.its.ac.id/ITS-Master-22103140001134/35060/muhammad-abdillah, pada Tanggal 22 Oktober 2015.

Akbar, A.L., Yudhistira, Novanto., dan Cholissodin, Imam., (2014). "Implementasi Algoritma SVM Untuk Mengetahui Tingkat Resiko Penyakit Stroke. Program Studi Teknik Informatika.

Batista, G.E.A.P.A., Bazzan, A.L.C. dan Monard, M.C, (2003). "Balancing Training Data for Automated Annotation of Keyword: a Cese study". *Proceedings of the second Brazilian Workshop Bioinformatics.* Diakses dari http://www.icmc.usp.br/~gbatista/files/wob2003.pdf, pada Tanggal 16 Oktober 2015).

Batista, G.E.A.P.A., Ronaldo, C. Prati dan Carolina, M. M. (2004). "A Study of The Behaviour of Several Methods for Balancing Machine Learning Training Data". *Sigkdd Explorations.* Vol 6, Issue 1, Hal 20.

Bhavsar, Hetal dan A. Ganatra. (2012), " Variation of Support Vector Machine Classification Technique : A survey", *International Journal of Advanced Computer Research* (ISSN (print) : 2249-7277, ISSN (online) : 2277-7970), Vol. 2, No. 4. Issue. 6 Desember 2012.

Breiman, L., Friedman, J., Olshen, R. dan Stone, C. (1984), *Classification and Regression Trees*, Wadsworth International Group.

Belhumeur, J. P. Hespanha, D. J. Kriegman, (1997). "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711–720, 1997.

Boyd, S., dan Vandenberghe, L. (2004), *Convex Optimization*. Cambridge University Press, Cambridge, USA

Burges C, (1998), "A Tutorial On Support Vector Machine for Pattern Recognition". *Data Mining and Knowledge Discovery*, Vol. 2, No. 2, Hal.955-974.

Chawla, N. V, Bowyer, K. W, Hall, L. O dan Kegelmeyer, W. P, (2002), "SMOTE:Synthetic Minority Oversampling Technique", *Journal of Artificial Intelligence Research*, Vol. 16, Hal.321-357.

Chawla, N. V. (2003), "C4.5 and Imbalanced Data set: Investigating the effect of Sampling Method, Probabilistic Estimate and Decision Tree Structure", In ICML Work Workshop on Learning from Imbalanced Data Set, Washington, D.C.

Chen, P.-H., C.-J.Lin dan B. Scholkopf. (2005), "A Tutuorial on v-Support Vector Machines Applied Stochastic Model in Business and Industry, Vol 21, Hal. 111-136.

Cheng, Hui-Ling., B. Yang, J.Liu dan D.Y.Liu, (2011). "A Support Vector Machine Classifier with rough set based feature selection for breast cancer diagnosis". *Expert System with Application*, Vol. 38, No 7, Hal 9014-9022.

Choi, J. M,(2010), "A Selective Sampling Method for Imbalanced Data Learning on Support Vector Machines",*Graduate Theses and Dissertations*, Paper 11529.

Cortez, C dan V. Vapnik (1995), "Support Vector Networks", *Machine Learning*, Vol. 20, No. 3, Hal. 273–297.

Daniel, W.W. (1989). *Statistika Nonparametrik Terapan*. PT Gramedia, Jakarta.

Estabrooks, A., Jo, T dan Japkowicz, N, (2004). " A Multiple Resampling Method for Learning from Imbalanced Dataset". *Journal Computational Intelligence*. Vol 20, Hal 18-36.

Gaudio, R.D., Batista, G., dan Branco, A., (2013) . "Coping with highly imbalanced dataset: A case study with definition extraction in a multilingual setting". *Natural Languange Engineering*, Hal 1-33, Cambridge University Press 2013. doi:10.1017/S1351324912000381.

Gunn, Steve. (1998), *Support Vector Machines for Clasification and Regression*, Technical Report, ISIS.

Guo, J.,Yi, Ping.,Wang, R., Ye, Qiaolin.,Zhao, Chunxia., (2014). "Feature Selection for Least Sqaure Projection Twin Support Vector Machine". *Neurocomputing*. Vol. 14, Hal. 174-183.

Haerdle, WK, Prastyo, DD, and Hafner, CM. (2014)."Support Vector Machines with Evolutionary Model Selection for Default Prediction," in *The Oxford Handbook of Applied Nonparametric and Semiparametric Econometrics and Statistics*, eds. Racine, JS, Su, L, and Ullah, A, Oxford University Press, 346-373.

Hair. J.F. (1995). *Multivariate Data Analisis*, Prentice-Hall International INC, U.S.A-Mexico-Canada

Han, J dan Kamber, M. (2001), *Data Mining Concept and Tehniques,* USA, Academic Press.

Han, J.,Kamber, M & Jian Pei. (2006). *Data Mining: Consepts and Tecniques (3nd ed)*, Morgan Kaufmaan, San Fransisco.

He, H dan E. Garcia, (2009), "Learning from imbalanced data", *IEEE Transactions on Knowledge an Data Engineering,* Vol. 21, No. 9, Hal 1263-1284

Hsu, C.W dan Lin, C.J, (2002), "A Comparison of Methods for Multiclass Support Vector Machines", *IEEE Trans. Neural Netw*, Vol. 13, No. 2, Hal.415–425.

Hsu, C.W, Chang, C.C., dan Lin, C.J, (2004), "A Practical Guide to Support Vector Classification", Department of Computer Scinece an Information Engineering, National Taiwan University.

Huang, C.M., Lee, Y.J., Lin D.K.J., dan Huang, S.Y. (2007), Model selection for support vector machine via uniform design, *Computational Statistics and Data Analysis, Vol. 52. hal. 335-346.*

Japkowicz., N. dan S. Stephen. (2002), "The Class Imbalanced Problem: A Systematic Study, *Intelligent Data Analysis*, Vol. 6 N0. 5, Hal.429-449.

Kennedy, J dan Eberhart, R.C, (1995), "Particle Swarm Optimization", *Proceedings of IEEE International Conference on Neural Network*, Piscataway, NJ, Hal. 1942-1948.

Kennedy J, Eberhart RC, dan Shi Y, (2001), *Swarm Intelligence*. Morgan Kaufman, CA.

Kubat, M dan Matwin, S, (1997). Addressing The Curse of Imbalanced Training Set : One Sided Selection, 14 th International Conference on Machine Learning Nashville, TN, USA,pp.179-186.

Kubat, M dan Matwin, S dan Holte, R., (1998). "Machine Learning for the detection of oil Spills in satellite Radar Images". *Journal Machine Learning*, Vol 30, Hal 195-215.

Lee, M.C dan To, Chang.,(2010), "Comparison of Support Vector Machine and Back Propagation Neural Network in Evaluating the Enterprise Financial Distress", *International Journal of Artificial& Applications (IJAIA)*, Vol.1, No.3, (July, 2010).

Lewis, D dan Carlett, J, (1994). "Heterogeneous Uncertainly Sampling for Supervised Learning, In Cohen, W.W. dan Hirsh, H. (Eds), Proceedings of ICML-94, 11 th International Conference on Machine Learning, Hal 148-156, San Fransisco, Morgan Kaufmann.

Ling, C, X dan Li, C., (1998). "Data mining for Direct Marketing Problem and Solution, Proceedings of the 4 th International Conference on Knowledge Discovery and Data Mining, New York, USA, hal 73-80.

Mercer, J. (1909), "Foundations of Positive and Negative Type, and Their Connection with the Theory of Integral Equations", *Philosophical Transactions of the Royal Society of London*, Vol. 25, Hal. 3-23.

Mirjalili S dan Hashim SZM, (2010), "A New Hybrid PSOGSA Algorithm for Function Optimization", *International Conference on Computer and Information Application (ICCIA)*.

Morisson, D. (2005). Multivariate Statistical Methods (Second Edition). The Wharton School University Of Pennsylvania, United States of America.

Newton I, (1729), "In experimental philosophy particular propositions are inferred from the phenomena and afterwards rendered general by induction", *3rd ed.: Andrew Motte's English translation published*, Vol. 2.

Novianti, A. Furina., Purnami, W. Santi. (2012), "Analisis Diagnosis Pasien Kanker Payudara Menggunakan Regresi Logistik dan Support Vector Machine (SVM) Berdasarkan Hasil Mammografi", *Jurnal SAINS dan SENI ITS,* Vol.1, No.1. (Sept. 2012) ISSN : 2301-928X.

Priya, R dan P. Aruna, (2012), "SVM and Neural Network based Diagnosis of Diabeteic Renithopathy", *International Journal of Computer Applications* (0975-8875). Vol. 41.No.1 (Maret, 2012).

Rahman, Farizi., Purnami, W. Santi. (2012), Perbandingan Klasifikasi Tingkat Keganasan Breast Cancer Dengan Menggunakan Regresi Logistik Ordinal Dan Support Vector Machine (SVM), Jurnal SAINS dan Seni ITS, Vol.1, No.1, (September 2012) ISSN : 2301-928X.

Robandi, I dan Prasetyo Gusti, R.A,(2008),*Peramalan Beban Jangka Pendek Untuk Hari-hari Libur Dengan Metode Support Vector Machine*, Tugas Akhir, ITS, Surabaya.

Sain, Hartayuni. (2013), *Combine sampling Support Vector Machine Untuk Klasifikasi Data Imbalanced*, Tesis, Statistika-FMIPA ITS, Surabaya.

Santosa, B, (2007), *Data Mining: Teknik Pemanfaatan Data Untuk Keperluan Bisnis, Teori dan Aplikasi*, Graha Ilmu.

Sastrawan, A.S., Baizal, Z.K. A., Bijaksana, M. A., (2010). "Analisis Pengaruh Metode Combine Sampling Dalam Churn Prediction Untuk Perusahaan Telekomunikasi". *Seminar Nasional Informatika 2010 (SeminasIF 2010)*,UPN "Veteran" Yogyakarta, 22 Mei 2010, ISSN : 1979-2328, Hal A.14-A.22.

Scholkopf, B dan A. Smola. (2002), *Learning with Kernel :Support Vector Machines, Regulerizatiom, Optimization, and Beyond*, Cambridge, MA : MIT Press

Sevita IA, Purnami SW, dan Wulandari SP, (2012), "Klasifikasi Pasien Hasil Pap Smear Test sebagai Upaya Pendeteksian Awal Upaya Penanganan Dini pada Penyakit Kanker Serviks di RS. "X" Surabaya dengan Metode Bagging Logistic Regression", *Jurnal SAINS dan SENI ITS,* Vol.1, No.1.

Solberg, A dan Solberg, R, (1996),"A Large-Scale Evaluationof Features for Automatic Detection of Oil Spills in ERS SAR Images",*InInternational Geoscience and Remote Sensing Symposium*, Hal. 1484–1486, Lincoln, NE.

Suykens, J. A. K., & Vandewalle, J. (Eds.) (1998). Nonlinear Modeling: Advanced Black-Box Techniques. Boston: Kluwer Academic Publishers.

Suykens, J. A. K., & Vandewalle, J. (1999a). "Training multilayer perceptron classifiers based on a modified support vector method". *IEEE Transactions on Neural Networks*, 10, 907–912.

Suykens, J. A. K.,dan Vandewalle, J. (1999b). "Least squares support vector machine classifiers". *Neural Processing Letters*, 9, 293–300.

Suykens, J. A. K., dan Vandewalle, J. (1999c). "Multiclass least squares support vector machines". In Proc. of the Int. Joint *Conf. on Neural Networks (IJCNN'99)*, Washington, DC.

Tan, P. N., Steinbach, M., dan Kumar, V. (2006).*Introduction to Data Mining (4th ed.*), Pearson Addison Wesley, Boston.

Tomek, I., (1998). "Two Modification of CNN". *IEEE Transactions on System Man and Communications*, SMC-6: 769-772, 1976.

Trapsilasiwi, R.K., (2013). *Klasifikasi Multiclass Untuk Imbalanced Data Menggunakan SMOTE Least Sqaure Support Vector Machine*, Tesis, Statistika FMIPA-ITS, Surabaya.

Vapnik, V., (1998), *The Nature of Statistical Learning, second ed*., Springer, New York.

Wu, G dan Chang, E., (2003). Class-Boundary Alignment for imbalanced Dataset Learning, In ICML 2003 Workshoop on Learning from Imbalanced Dataset II, Washington, DC.

Yohannes, Y dan Webb, P. (1999), *Classification and Regression Trees, A user Manual for Identfying of Vulnerability to famine and chronic food Insecurity,* Microcomputers in Policy Research International Food Policy Research Institute, Washington, D.C, USA.

Zheng, H.B., Liao, R.J., Grzybowski, S dan Yang, L.J.,(2011). "Fault diagnosis of power transformers using multi-class least square support vector machines classifier with particle swarm optimisation". IET Elect. Power Appl. Vol 5, Iss 9, Hal 691-696, doi : 10.1049/iet-epa. 2010. 0298.